

CHAPTER 1

INTRODUCTION

1.1 Overview

Wikipedia defines Big Data as "a collection of data sets so large and complex that it becomes difficult to process using the available database management tools. The challenges include how to capture, curate, store, search, share, analyze and visualize Big Data". In today's environment, we have access to more types of data. These data sources include online transactions, social networking activities, mobile device services, internet gaming etc.

Big Data is a collection of data sets that are large and complex in nature. They constitute both structured and unstructured data that grow large so fast that they are not manageable by traditional relational database systems or conventional statistical tools. Big Data is defined as any kind of data source that has at least three shared characteristics:

- Extremely large Volumes of data
- Extremely high Velocity of data
- Extremely wide Variety of data

According to Big Data: Concepts, Methodologies, Tools, and Applications, Volume I by Information Resources Management Association (IRMA), "organizations today are at the tipping point in terms of managing data. Data sources are ever expanding. Data from Facebook, Twitter, YouTube, Google etc., are to grow 60X in the next 10 years. Over 2.6 exabytes of data is generated every day. Some of the sources of huge volume of data are:

1. A typical large stock exchange captures more than 1 TB of data every day.
2. There are over 6 billion mobile phones in the world which are producing enormous amount of data on daily basis.
3. YouTube users upload more than 48 hours of video every minute.
4. Large social networks such as Twitter and Facebook capture more than 10 TB of data daily.

6. There are more than 30 million networked sensors in the world which further produces TBs of data every day. "

Structured and semi-structured formats have some limitations with respect to handling large quantities of data. Hence, in order to manage the data in the Big Data world, new emerging approaches are required, including document, graph, columnar, and geospatial database architectures. Collectively, these are referred to as NoSQL, or not only SQL, databases. In essence the data architectures need to be mapped to the types of transactions. Doing so will help to ensure the right data is available when you need it.

1.2 Hadoop

As organizations are getting flooded with massive amount of raw data, the challenge here is that traditional tools are poorly equipped to deal with the scale and complexity of such kind of data. That's where Hadoop comes in. Hadoop is well suited to meet many Big Data challenges, especially with high volumes of data and data with a variety of structures.

At its core, Hadoop is a framework for storing data on large clusters of commodity hardware — everyday computer hardware that is affordable and easily available — and running applications against that data. A cluster is a group of interconnected computers (known as nodes) that can work together on the same problem. Using networks of affordable compute resources to acquire business insight is the key value proposition of Hadoop.

Hadoop consists of two main components:

1. A distributed processing framework named MapReduce (which is now supported by a component called YARN (Yet another Resource Negotiator) and
2. A distributed file system known as the Hadoop Distributed File System, or HDFS.

In Hadoop you can do any kind any kind of aggregation of data whether it is one month old data or one-year-old data. Hadoop provides a mechanism called MapReduce model to do distributed processing of large data which internally takes care of data even if one machine goes down.

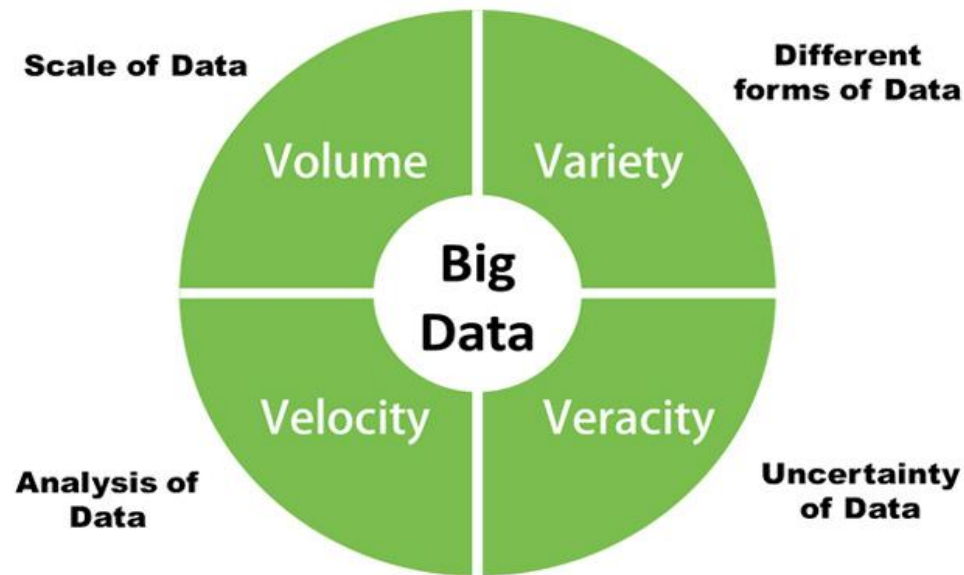


Fig 1.1 Hadoop Characteristics

1.3 Hadoop Ecosystem

Hadoop is a shared nothing system where each node acts independently throughout the system. A framework where a piece of work is divided among several parallel MapReduce task. Each task operated independently on cheap commodity servers. This enables businesses to generate values from data that was previously considered too expensive to be stored and processed in a traditional data warehouse or OLTP (Online Transaction Processing) environment.

In the old paradigm, companies would use a traditional enterprise data warehouse system and would buy the biggest data warehouse they could afford and store the data on a single machine. However, with the increasing amount of data, this approach is no longer affordable nor practical.

Some of the components of Hadoop ecosystem are HDFS (Hadoop Distributed File System), MapReduce, Yarn, Hive and Hbase. Hadoop has two core components. 'Storage' part to store the data and 'Processing' part to process the data. The storage part is called 'HDFS' and the processing part is called as 'YARN'.

Hadoop Ecosystem

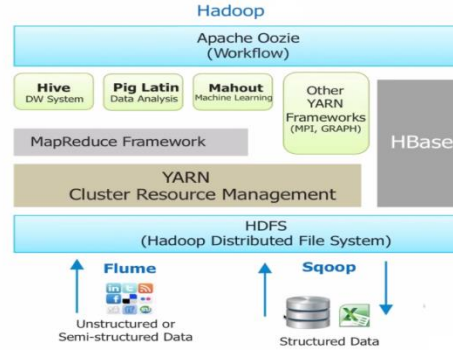


Fig 1.2 Hadoop Ecosystem

1.4 Sketching Out the HDFS Architecture

1.4.1 Storage Component: Hadoop Distributed File System (HDFS)

As stated above, the Hadoop Distributed File System (HDFS) is the storage component of the core Hadoop Infrastructure. HDFS provides a distributed architecture for extremely large scale storage, which can easily be extended by scaling out. It is important to mention the difference between scale up and scale out. In its initial days, Google was facing challenges to store and process not only all the pages on the internet but also its users' web log data. At that time, Google was using scale up architecture model where you can increase the system capacity by adding CPU cores, RAM etc to the existing server. But such kind of model had not only been expensive but also had structural limitations. So instead, Google engineers implemented Scale out architecture model by using a cluster of smaller servers which can be further scaled out if they require more power and capacity. Google File System (GFS) was developed based on this architectural model. HDFS is designed based on a similar concept.

The core concept of HDFS is that it can be made up of dozens, hundreds, or even thousands of individual computers, where the system's files are stored in directly attached disk drives. Each of these individual computers is a self-contained server with its own memory, CPU, disk storage, and installed operating system (typically Linux, though Windows is also supported). Technically speaking, HDFS is a user-space-level file

system because it lives on top of the file systems that are installed on all individual computers that make up the Hadoop cluster.

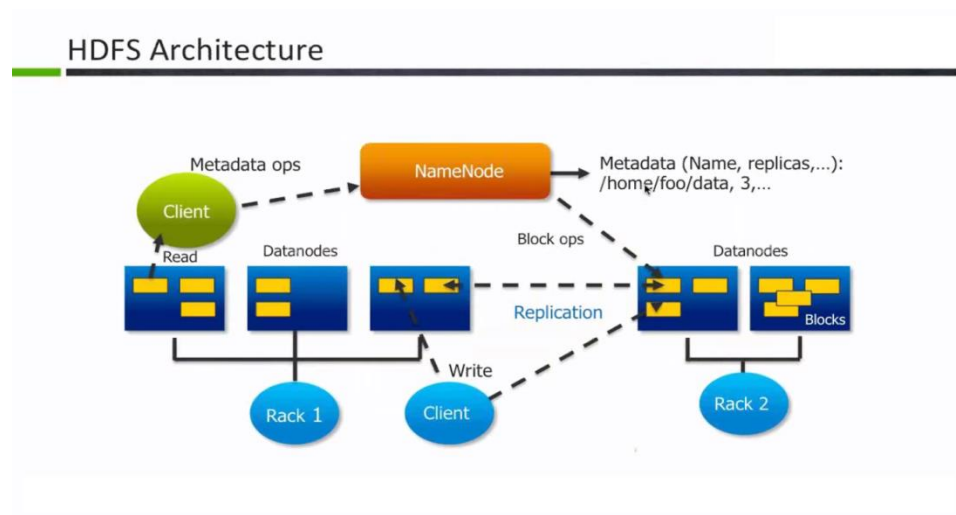


Fig 1.3

The above figure shows that a Hadoop cluster is made up of two classes of servers: slave nodes, where the data is stored and processed and master nodes, which govern the management of the Hadoop cluster. On each of the master nodes and slave nodes, HDFS runs special services and stores raw data to capture the state of the file system. In the case of the slave nodes, the raw data consists of the blocks stored on the node, and with the master nodes, the raw data consists of metadata that maps data blocks to the files stored in HDFS.

HDFS is a system that allows multiple commodity machines to store data from a single source. HDFS consists of a NameNode and a DataNode. HDFS operates as master slave architecture as opposed to peer to peer architecture. NameNode serves as the master component while the DataNode serves as a slave component. NameNode comprises of only the Meta data information of HDFS that is the blocks of data that are present on the Data Node

- How many times the data file has been replicated?
- When does the NameNode start?

- How many DataNodes constitute a NameNode, capacity of the NameNode and space utilization?
- The DataNode comprises of data processing, all the processing data that is stored on the DataNode and deployed on each machine.
- The actual storage of the files being processed and serving read and write request for the clients.

In the earlier versions of Hadoop there was only one NameNode attached to the DataNode which was a single point of failure. Hadoop version 2.x provides multiple NameNode where secondary NameNode can take over in the event of a primary NameNode failure. Secondary NameNode is responsible for performing periodic checkpoints in the event of a primary NameNode failure. You can start secondary NameNode by providing checkpoints that provide high availability within HDFS.

But what happens if one of these four machines fails? HDFS creates a self-healing architecture by replicating the same data across multiple nodes. So it can process the data in a high availability environment. For example, if we have three DataNodes and one NameNode, the data is transferred from the client environment into HDFS DataNode. The replication factor defines the number of times a data block is replicated in a clustered environment. Let's say we have a file that is split into two data blocks across three DataNodes. If we are processing these files to a three DataNode cluster and we set the replication factor to three. If one of the nodes fails, the data from the failed nodes is redistributed among the remaining active nodes and the other nodes will complete the processing function.

1.4.2 Processing Component: Yet Another Resource Negotiator (YARN)

YARN (Yet Another Resource Negotiator) is a resource manager that identifies on which machine a particular task is going to be executed. The actual processing of the task or program will be done by Node Manager. In Hadoop 2.2, YARN augments the MapReduce platform and serves as the Hadoop operating system. Hadoop 2.2 separates the resource management function from data processing allowing greater flexibility. This way MapReduce only performs data processing while resource management is isolated in

YARN. Being the primary resource manager in HDFS, YARN enables enterprises to store data in a single place and interact with it in multiple ways with consistent levels of service. In Hadoop 1.0 the NameNode used job tracker and the DataNode used task tracker to manage resources. In Hadoop 2.x, YARN splits up into two major functionalities of the job tracker - the resource management and job scheduling. The client reports to the resource manager and the resource manager allocates resources to jobs using the resource container, Node Manager and app master. The resource container splits memory, CPU, network bandwidth among other hardware constraints into a single unit. The Node Manager receives updates from the resource containers which communicate with the app master. The Node Manager is the framework for containers, resource monitoring and for reporting data to the resource manager and scheduler.

CHAPTER 2

SOFTWARE AND HARDWARE REQUIREMENTS

2.1 Software Requirements

Operating System	: Windows 8/10 and Linux
To run other OS on window	: Virtual Machine
To run Ubuntu terminal on window	: Putty
To transfer file from window to ubuntu	: WinScp
Data Bases	: Oracle/Mysql

2.2 Hardware Requirements

Processor	: Intel i3 or above
Hard Disk	: 100 GB or as per requirement
RAM	: 4 GB min
Network	: 1 GB Ethernet

CHAPTER 3

SOFTWARE REQUIREMENT ANALYSIS

Hadoop is an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation. Hadoop makes it possible to run applications on systems with thousands of commodity hardware nodes, and to handle thousands of terabytes of data. Its distributed file system facilitates rapid data transfer rates among nodes and allows the system to continue operating in case of a node failure. This approach lowers the risk of catastrophic system failure and unexpected data loss, even if a significant number of nodes become inoperative. Consequently, Hadoop quickly emerged as a foundation for big dataprocessing tasks, such as scientific analytics, business and sales planning, and processing enormous volumes of sensor data, including from internet of things sensors. Hadoop was created by computer scientists Doug Cutting and Mike Cafarella in 2006 to support distribution for the Nutchsearchengine. It was inspired by Google's MapReduce, a software framework in which an application is broken down into numerous small parts. Any of these parts, which are also called fragments or blocks, can be run on any node in the cluster. After years of development within the open source community, Hadoop 1.0 became publicly available in November 2012 as part of the Apache project sponsored by the Apache Software Foundation.

3.1 Tools of Hadoop

3.1.1 Apache Flume

A tool used to collect, aggregate and move huge amounts of streaming data into HDFS. Flume's high-level architecture is built on a streamlined codebase that is easy to use and extend. The project is highly reliable, without the risk of data loss. Flume also supports dynamic reconfiguration without the need for a restart, which reduces downtime for its agents.

Flume components interact in the following way:

1. A flow in Flume starts from the Client.

2. The Client transmits the Event to a Source operating within the Agent.
3. The Source receiving this Event then delivers it to one or more Channels.
4. One or more Sinks operating within the same Agent drains these Channels.
5. Channels decouple the ingestion rate from drain rate using the familiar producer-consumer model of data exchange.
6. When spikes in client side activity cause data to be generated faster than can be handled by the provisioned destination capacity can handle, the Channel size increases. This allows sources to continue normal operation for the duration of the spike.
7. The Sink of one Agent can be chained to the Source of another Agent. This chaining enables the creation of complex data flow topologies. Because Flume's distributed architecture requires no central coordination point. Each agent runs independently of others with no inherent single point of failure, and Flume can easily scale horizontally.

3.1.2 Apache HBase

An open source, non-relational, distributed database. HBase is designed to support high table-update rates and to scale out horizontally in distributed compute clusters. Its focus on scale enables it to support very large database tables -- for example, ones containing billions of rows and millions of columns. Currently, one of the most prominent uses of HBase is as a structured data handler for Facebook's basic messaging infrastructure. HBase is known for providing strong data consistency on reads and writes, which distinguishes it from other NoSQL databases.

3.1.3 Apache Hive

A data warehouse that provides data summarization, query and analysis. Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop. Traditional SQL queries must be implemented in the MapReduce Java API to execute SQL applications and queries over distributed data. Hive provides the necessary SQL abstraction to integrate SQL-like queries (HiveQL) into the underlying Java without the need to implement queries in the low-level Java API. Since most data warehousing applications work with SQL-based querying languages, Hive aids portability of SQL-based applications to Hadoop. While initially developed by Facebook,

Apache Hive is used and developed by other companies such as Netflix and the Financial Industry Regulatory Authority (FINRA).

3.1.4 Cloudera Impala

A massively parallel processing database for Hadoop, originally created by the software company Cloudera, but now released as open source software. Impala is promoted for analysts and data scientists to perform analytics on data stored in Hadoop via SQL or business intelligence tools. The result is that large-scale data processing (via MapReduce) and interactive queries can be done on the same system using the same data and metadata – removing the need to migrate data sets into specialized systems and/or proprietary formats simply to perform analysis.

Features includes

Supports HDFS and Apache HBase storage,

- Reads Hadoop file formats, including text, LZO, Sequence File, Avro, RCFile, and Parquet,
- Supports Hadoop security (Kerberos authentication),
- Fine-grained, role-based authorization with Apache Sentry,
- Uses metadata, ODBC driver, and SQL syntax from Apache Hive.

3.1.5 Apache Oozie.

A server-based workflow scheduling system to manage Hadoop jobs. Workflows in Oozie are defined as a collection of control flow and action nodes in a directed acyclic graph. Control flow nodes define the beginning and the end of a workflow (start, end, and failure nodes) as well as a mechanism to control the workflow execution path (decision, fork, and join nodes). Action nodes are the mechanism by which a workflow triggers the execution of a computation/processing task. Oozie provides support for different types of actions including Hadoop MapReduce, Hadoop distributed file system operations, Pig, SSH, and email. Oozie can also be extended to support additional types of actions.

3.1.6 Apache Phoenix

An open source, massively parallel processing, relational database engine for Hadoop that is based on Apache HBase. Apache Phoenix takes your SQL query, compiles it into a series of HBase scans and orchestrates the running of those scans to produce regular JDBC result sets. Direct use of the HBase API, along with coprocessors and custom filters, results in performance on the order of milliseconds for small queries, or seconds for tens of millions of rows.

3.1.7 Apache Pig.

A high-level platform for creating programs that run on Hadoop. Pig can execute its Hadoop jobs in MapReduce, Apache Tez, or Apache Spark. Pig Latin abstracts the programming from the JavaMapReduce idiom into a notation which makes MapReduce programming high level, similar to that of SQL for relational database management systems. Pig Latin can be extended using user-defined functions (UDFs) which the user can write in Java, Python, JavaScript, Ruby or Groovy and then call directly from the language.

3.1.8 Apache Sqoop.

A tool to transfer bulk data between Hadoop and structured data stores, such as relational databases. Sqoop supports incremental loads of a single table or a freeform SQL query as well as saved jobs which can be run multiple times to import updates made to a database since the last import. Imports can also be used to populate tables in Hive or HBase. Exports can be used to put data from Hadoop into a relational database. Sqoop got the name from sql+hadoop. Sqoop became a top-level Apache project in March 2012.

3.1.9 Apache Spark.

A fast engine for big data processing capable of streaming and supporting SQL, machine learning and graph processing. Apache Spark is an open-source cluster-computing framework. Originally developed at the University of California, Berkeley's AMPLab, the Spark codebase was later donated to the Apache Software Foundation, which has maintained it since. Spark provides an interface for programming entire clusters with implicit data parallelism and fault-tolerance.

3.1.10 Apache Storm.

An open source data processing system Apache Storm is a free and open sourced distributed Real-time computation system. Storm makes it easy to reliably process unbounded streams of data, doing for Real-time processing what Hadoop did for batch processing. Storm is simple, can be used with any programming language, and is a lot of fun to use!

3.1.11 Apache ZooKeeper.

An open source configuration, synchronization and naming registry service for large distributed systems. ZooKeeper's architecture supports high availability through redundant services. The clients can thus ask another ZooKeeper leader if the first fails to answer. ZooKeeper nodes store their data in a hierarchical name space, much like a file system or a tree data structure. Clients can read from and write to the nodes and in this way have a shared configuration service. ZooKeeper can be viewed as an atomic broadcast system, through which updates are totally ordered. The ZooKeeper Atomic Broadcast (ZAB) protocol is the core of the system.

3.1.12 Map Reduce.

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

3.2 Apache Pig

3.2.1 Introduction of Pig

In Map Reduce framework, programs need to be translated into a series of Map and Reduce stages. However, this is not a programming model which data analysts are familiar with. So, in order to bridge this gap, an abstraction called Pig was built on top of Hadoop. Pig is a high level programming language useful for analyzing large data sets.

Pig was a result of development effort at Yahoo! Pig enables people to focus more on analyzing bulk data sets and to spend less time in writing Map-Reduce programs.

Similar to Pigs, who eat anything, the Pig programming language is designed to work upon any kind of data. That's why the name, Pig!



Fig 3.1 Pig Symbol

Pig consists of two components:

1. Pig Latin, which is a language
2. Runtime environment, for running Pig Latin programs.

A Pig Latin program consists of a series of operations or transformations which are applied to the input data to produce output. These operations describe a data flow which is translated into an executable representation, by Pig execution environment. Underneath, results of these transformations are series of MapReduce jobs which a programmer is unaware of. So, in a way, Pig allows programmer to focus on data rather than the nature of execution.

Pig Latin is a relatively stiffened language which uses familiar keywords from data processing e.g., Join, Group and Filter.

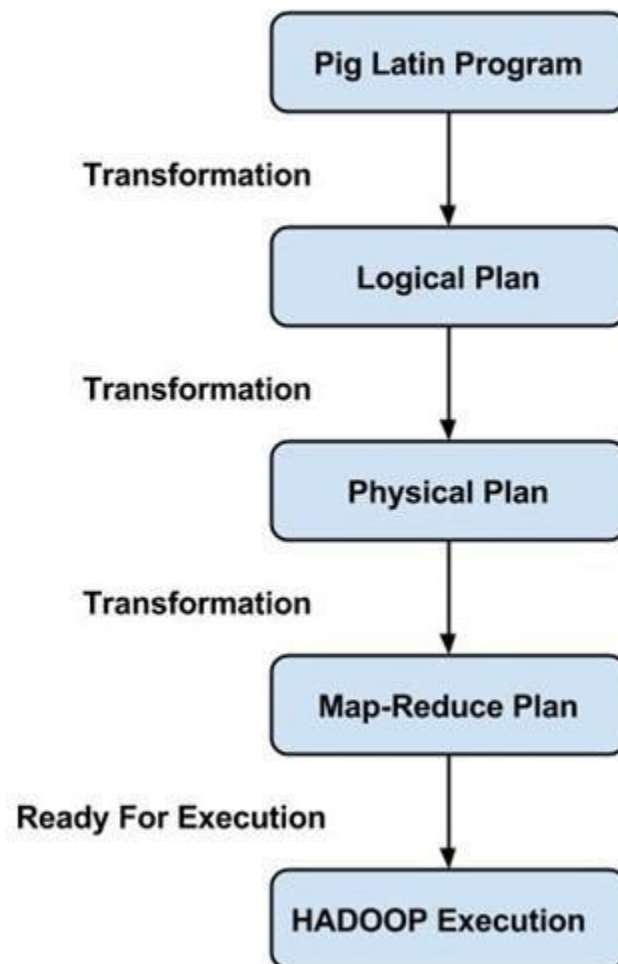


Fig 3.2 Pig Cycle

Execution modes

Pig has two execution modes:

1. **Local mode:** In this mode, Pig runs in a single JVM and makes use of local file system. This mode is suitable only for analysis of small data sets using Pig
2. **Map Reduce mode:** In this mode, queries written in Pig Latin are translated into MapReduce jobs and are run on a Hadoop cluster (cluster may be pseudo or fully distributed). MapReduce mode with fully distributed cluster is useful of running Pig on large data sets.

3.2.2 Installation of Pig

Below are the steps for Apache Pig Installation on Linux (ubuntu/centos/windows using Linux VM). I am using Ubuntu 16.04 in below setup.

Step 1: Download Pig tar file.

Command: we get <http://www-us.apache.org/dist/pig/pig-0.16.0/pig-0.16.0.tar.gz>



```
edureka@localhost:~$ wget http://www-us.apache.org/dist/pig/latest/pig-0.16.0.tar.gz
--2016-11-18 17:46:31-- http://www-us.apache.org/dist/pig/latest/pig-0.16.0.tar.gz
Resolving www-us.apache.org (www-us.apache.org)... 140.211.11.105
Connecting to www-us.apache.org (www-us.apache.org)|140.211.11.105|:80..
. connected.
HTTP request sent, awaiting response... 200 OK
Length: 177279333 (169M) [application/x-gzip]
Saving to: 'pig-0.16.0.tar.gz'

pig-0.16.0.tar.gz  2%[          ]  4.80M  149KB/s  eta 9m 0s
```

Fig 3.3 Command shell

Step 2: Extract the tar file using tar command. In below tar command, x means extract an archive file, zmeans filter an archive through gzip, f means filename of an archive file.

Command: tar -xzf pig-0.16.0.tar.gz

Command: ls


```

edureka@localhost:~$ tar -xzf pig-0.16.0.tar.gz
edureka@localhost:~$ ls
apache-hive-2.1.0-bin      jdk-8u101-linux-i586.tar.gz
apache-hive-2.1.0-bin.tar.gz Music
derby.log                 Pictures
Desktop                   pig-0.16.0
Documents                 pig-0.16.0.tar.gz
Downloads                 Public
examples.desktop          Templates
hadoop-2.7.3              Videos
hadoop-2.7.3.tar.gz
edureka@localhost:~$

```

Fig 3.4 Command shell

Step 3: Edit the “.bashrc” file to update the environment variables of Apache Pig. We are setting it so that we can access pig from any directory, we need not go to pig directory to execute pig commands. Also, if any other application is looking for Pig, it will get to know the path of Apache Pig from this file.

Command: `sudoedit .bashrc`

Add the following at the end of the file:

```

# Set PIG_HOME

export PIG_HOME=/home/edureka/pig-0.16.0
export PATH=$PATH:/home/edureka/pig-0.16.0/bin
export PIG_CLASSPATH=$HADOOP_CONF_DIR

```

Also, make sure that hadoop path is also set.

Run below command to make the changes get updated in same terminal.

Command: `source .bashrc`

Step 4: Check pig version. This is to test that Apache Pig got installed correctly. In case, you don't get the Apache Pig version, you need to verify if you have followed the above steps correctly.

Command: `pig -version`

```

edureka@localhost:~$ source .bashrc
edureka@localhost:~$ pig -version
Apache Pig version 0.16.0 (r1746530)
compiled Jun 01 2016, 23:10:49

```

Fig 3.5 Command shell

Step 5: Check pig help to see all the pig command options.

Command: pig -help

```
edureka@localhost:~$ pig -help

Apache Pig version 0.16.0 (r1746530)
compiled Jun 01 2016, 23:10:49

USAGE: Pig [options] [-] : Run interactively in grunt shell.
      Pig [options] -e[execute] cmd [cmd ...] : Run cmd(s).
      Pig [options] [-f[file]] file : Run cmds found in file.
options include:
  -4, -log4jconf - Log4j configuration file, overrides log conf
  -b, -brief - Brief logging (no timestamps)
  -c, -check - Syntax check
  -d, -debug - Debug level, INFO is default
  -e, -execute - Commands to execute (within quotes)
  -f, -file - Path to the script to execute
  -g, -embedded - ScriptEngine classname or keyword for the ScriptEngine
```

Fig 3.6 Command shell

Step 6: Run Pig to start the grunt shell. Grunt shell is used to run Pig Latin scripts.

Command: pig

```
edureka@localhost:~$ pig
16/11/18 18:23:05 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
16/11/18 18:23:05 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
16/11/18 18:23:05 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2016-11-18 18:23:05,903 [main] INFO org.apache.pig.Main - Apache Pig version 0.16.0 (
r1746530) compiled Jun 01 2016, 23:10:49
2016-11-18 18:23:05,903 [main] INFO org.apache.pig.Main - Logging error messages to:
/home/edureka/pig_1479473585894.log
2016-11-18 18:23:06,035 [main] INFO org.apache.pig.impl.util.Utils - Default bootup f
ile /home/edureka/.pigbootup not found
2016-11-18 18:23:07,666 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable
to load native-hadoop library for your platform... using builtin-java classes where ap
plicable
2016-11-18 18:23:07,748 [main] INFO org.apache.hadoop.conf.Configuration.deprecation
- mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2016-11-18 18:23:07,748 [main] INFO org.apache.hadoop.conf.Configuration.deprecation
- fs.default.name is deprecated. Instead, use fs.defaultFS
2016-11-18 18:23:07,749 [main] INFO org.apache.pig.backend.hadoop.executionengine.HEx
ecutionEngine - Connecting to hadoop file system at: hdfs://localhost:9000
2016-11-18 18:23:09,138 [main] INFO org.apache.pig.PigServer - Pig Script ID for the
session: PIG-default-09da455b-2390-4800-91f8-9e642ee4ebfe
2016-11-18 18:23:09,139 [main] WARN org.apache.pig.PigServer - ATS is disabled since
yarn.timeline-service.enabled set to false
grunt> █
```

Fig 3.7 Command shell

CHAPTER 4

SOFTWARE DESIGN

4.1 YouTube Data Analyses

YouTube, owned by Google, is a video sharing website, where users can upload, watch and share videos with others. YouTube provides a forum for people to connect, inform, and inspire others across the globe and acts as a distribution platform for original content creators and advertisers large and small.

According to Statista.com (The Statistics Portal), "As of July 2016, more than 400 hours of video content were uploaded to YouTube every minute, a fourfold increase compared to only two years prior. The platform, which was created in 2006, has slowly become one of the most visited websites in the world and a global phenomenon. In the first quarter of 2016, more than 80 percent of global internet users had visited YouTube in the last month. In the United States, it is the second largest social media website after Facebook, accounting for over 22 percent of social media traffic. The rise in Smartphone and other mobile devices usage has also helped increase the consumption of YouTube videos on the go. As of mid 2016, approximately half of U.S. mobile users accessed YouTube via a mobile device, whether Smartphone or tablet computer."

While companies, musicians or film distributors might use YouTube as a form of free direct advertisement, YouTube has also become a launch pad for various products/services wherein large corporations reveal the first look of the product on YouTube, generate a buzz about their product, assess the market demand based upon likes and view counts and improve their product based upon customers' feedback. Hence the data points including View Counts, Likes, Votes, and Comments etc. become very critical for the companies so that they can do the analysis and understand the customers' sentiments about their product/services.

The main objective of this project is to show how companies can analyze YouTube data using YouTube API to make targeted real time and informed decisions. This project will help in understanding changing trends among people by analyzing YouTube data and fetching meaningful results. For example, when companies like Disney launch their new

movie trailers on YouTube, this application can help Disney in analyzing the reaction of people towards a specific movie trailer. This application can analyze how many people liked the trailers, in which country the trailer was liked the most, whether the comments posted on YouTube are generally positive, negative or neutral etc. This way management can take executive decisions how to spend their marketing budget in order to maximize their returns.

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. Hadoop is definitely the preferred framework to amount of data to make it usable. Hadoop is definitely the preferred framework to analyze the data of this magnitude.

4.2 Hadoop Data Analysis Technologies

While Hadoop provides the ability to collect data on HDFS (Hadoop Distributed File System), there are many applications available in the market (like MapReduce, Pig, Flume, Sqoop and Hive) that can be used to analyze the data.

Apache Pig.

Pig is a high level programming language useful for analyzing large data sets. Pig was a result of development effort at Yahoo!

Pig enables people to focus more on analyzing bulk data sets and to spend less time in writing Map-Reduce programs.

Pig components

Pig consists of two components:

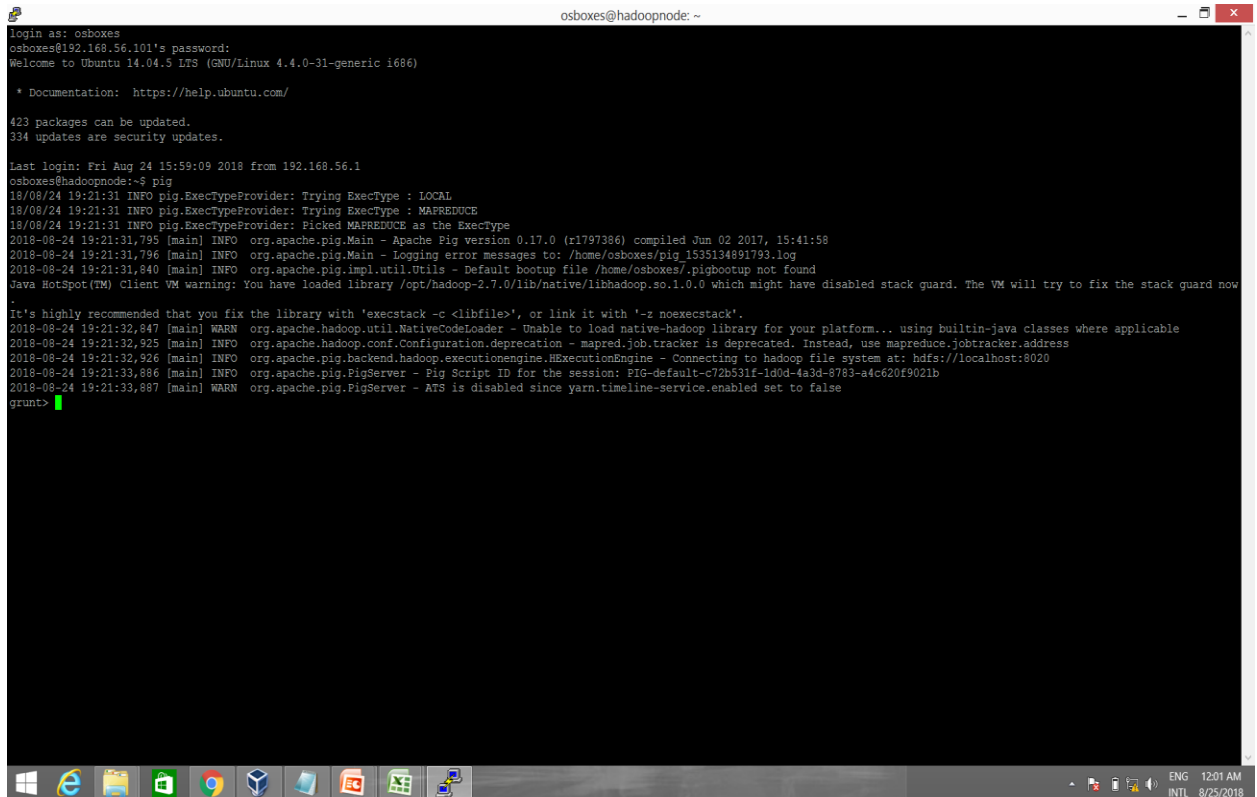
- Pig Latin, which is a language
- Runtime environment, for running Pig Latin programs.

A Pig Latin program consists of a series of operations or transformations which are applied to the input data to produce output. These operations describe a data flow which is translated into an executable representation, by Pig execution environment. Underneath, results of these transformations are series of MapReduce jobs

which a programmer is unaware of. So, in a way, Pig allows programmer to focus on data rather than the nature of execution.

Pig Latin is a relatively stiffened language which uses familiar keywords from data processing e.g., Join, Group and Filter.

Pig Shell



```
osboxes@hadoopnode: ~
login as: osboxes
osboxes@192.168.56.101's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)

 * Documentation:  https://help.ubuntu.com/

423 packages can be updated.
334 updates are security updates.

Last login: Fri Aug 24 15:59:09 2018 from 192.168.56.1
osboxes@hadoopnode:~$ pig
18/08/24 19:21:31 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
18/08/24 19:21:31 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
18/08/24 19:21:31 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2018-08-24 19:21:31,795 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58
2018-08-24 19:21:31,796 [main] INFO org.apache.pig.Main - Logging error messages to: /home/osboxes/pig/1535134891793.log
2018-08-24 19:21:31,840 [main] INFO org.apache.pig.impl.util.Utils - Default bootstrap file /home/osboxes7/pigbootstrap not found
Java HotSpot(TM) Client VM warning: You have loaded library /opt/hadoop-2.7.0/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'.
2018-08-24 19:21:32,847 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2018-08-24 19:21:32,925 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.job.tracker.address
2018-08-24 19:21:32,926 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:8020
2018-08-24 19:21:33,886 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-C72b531f-1d0d-4a3d-8783-a4c620f9021b
2018-08-24 19:21:33,887 [main] WARN org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
grunt>
```

Fig 4.1 Pig Shell

4.3 Dataset Schema

Column 1: Video id of 11 characters.

Column 2: uploader of the video

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

Column 4: Category of the video.

Column 6: 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 done on the videos.

Column Name	Data Type
video_id	chararray
Channel	chararray
Days	int
Category	chararray
Length	chararray
Views	int
Rating	float
no_rate	int
no_cmmnt	int

Table 4.1 Dataset schema

4.4 Analyzing factor

Here we compare these categories on the basis of their:-

- Count of a category.
- Average user rating(5 stars).
- Total no. of views.
- Maximum views on an individual video.
- No. of rating given to video per 1000.

CHAPTER 5

COMMANDS

Command to load data set in Pig

```
Grunt>youtube = load '/pig/youtubedata.txt' usingPigStorage('\t') AS
        (video_id:chararray, channel:chararray, days:int, category:chararra
        y, lenght:chararray, views:int, rating:float, no_rate:int,
        no_cmmnt:int);
```

```
Grunt>Dump youtube;
```

Description of load command

The Load Operator

You can load data into Apache Pig from the file system (HDFS/ Local) using LOAD operator of Pig Latin.

Syntax

The load statement consists of two parts divided by the “=” operator. On the left-hand side, we need to mention the name of the relation where we want to store the data, and on the right-hand side, we have to define how we store the data. Given below is the syntax of the Load operator.

```
Relation_name = LOAD 'Input file path' USING function as schema;
```

Where,

- relation_name – We have to mention the relation in which we want to store the data.
- Input file path – We have to mention the HDFS directory where the file is stored. (In MapReduce mode)
- function – We have to choose a function from the set of load functions provided by Apache Pig (BinStorage, JsonLoader, PigStorage, TextLoader).
- Schema – We have to define the schema of the data. We can define the required schema as follows –

```
(column1 : data type, column2 : data type, column3 : data type);
```

Grouping Command

```
Grunt > grp = GROUP youtube as category;
```

```
Grunt > dump grp;
```

Description Of GROUP

The GROUP operator is used to group the data in one or more relations. It collects the data having the same key.

Syntax

Given below is the syntax of the group operator.

```
grunt>Group_data = GROUP Relation_name BY age;
```

Count Command

```
Grunt >cnt = foreach grp generate group as grp,COUNT(category) as category;
```

```
Grunt > dump cnt;
```

Description of Foreach and COUNT

The FOREACH operator is used to generate specified data transformations based on the column data.

Syntax

Given below is the syntax of FOREACH operator.

```
grunt> Relation_name2 = FOREACH Relatin_name1 GENERATE (required data);
```

The COUNT() function of Pig Latin is used to get the number of elements in a bag. While counting the number of tuples in a bag, the COUNT() function ignores (will not count) the tuples having a NULL value in the FIRST FIELD.

- To get the global count value (total number of tuples in a bag), we need to perform a Group All operation, and calculate the count value using the COUNT() function.

- To get the count value of a group (Number of tuples in a group), we need to group it using the Group By operator and proceed with the count function.

Syntax

Given below is the syntax of the COUNT() function.

```
grunt> COUNT(expression)
```

Average Command

```
Grunt>yavg =foreach grp generate group as category,AVG(youtube.rating) as avg_rate;
```

```
Grunt> dump yavg;
```

Description of Average command

The Pig-Latin AVG() function is used to compute the average of the numerical values within a bag. While calculating the average value, the AVG() function ignores the NULL values.

- To get the global average value, we need to perform a Group All operation, and calculate the average value using the AVG() function.
- To get the average value of a group, we need to group it using the Group By operator and proceed with the average function.

Syntax

Given below is the syntax of the AVG() function.

```
grunt> AVG(expression)
```

Total no. of views command

```
Grunt>ysum =foreach grp generate group as category,SUM(youtube.views) as views;
```

```
Grunt> dump ysum;
```

Maximum views on an individual video command

```
Grunt>ymax = foreachgrp generate group as category,MAX(youtube.views) as max_views;
```

```
Grunt>dumpymax;
```

Description of sum command

You can use the SUM() function of Pig Latin to get the total of the numeric values of a column in a single-column bag. While computing the total, the SUM() function ignores the NULL values.

- To get the global sum value, we need to perform a Group All operation, and calculate the sum value using the SUM() function.
- To get the sum value of a group, we need to group it using the Group By operator and proceed with the sum function.

Syntax

Given below is the syntax of the SUM() function.

```
grunt> SUM(expression)
```

Number of ratings given for the video per 1000 command

```
Grunt> yavg1 =foreach grp generate group as category,AVG(youtube.no_rate) as  
avgrate;
```

```
Grunt> dump yavg1;
```

Description of Average command

The Pig-Latin AVG() function is used to compute the average of the numerical values within a bag. While calculating the average value, the AVG() function ignores the NULL values.

- To get the global average value, we need to perform a Group All operation, and calculate the average value using the AVG() function.
- To get the average value of a group, we need to group it using the Group By operator and proceed with the average function.

Syntax

Given below is the syntax of the AVG() function.

```
grunt> AVG(expression)
```

CHAPTER 6

TESTING

Testing is the process of evaluation a software item to detect differences between given input and expected output. Also, to assess the feature of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process.

6.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

6.2 Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

6.3 Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

6.4 System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

6.5 Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavourable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

6.6 Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

6.7 Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

6.8 Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

6.9 Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

6.10 Beta Testing

Beta testing is the testing which is done by end users, a team outside development, or publicly releasing full pre-version of the product which is known as beta version. The aim of beta testing is to cover unexpected errors. It falls under the class of black box testing.

CHAPTER 7

OUTPUT SCREENS

[illegible]

Fig 7.1 YouTube Dataset

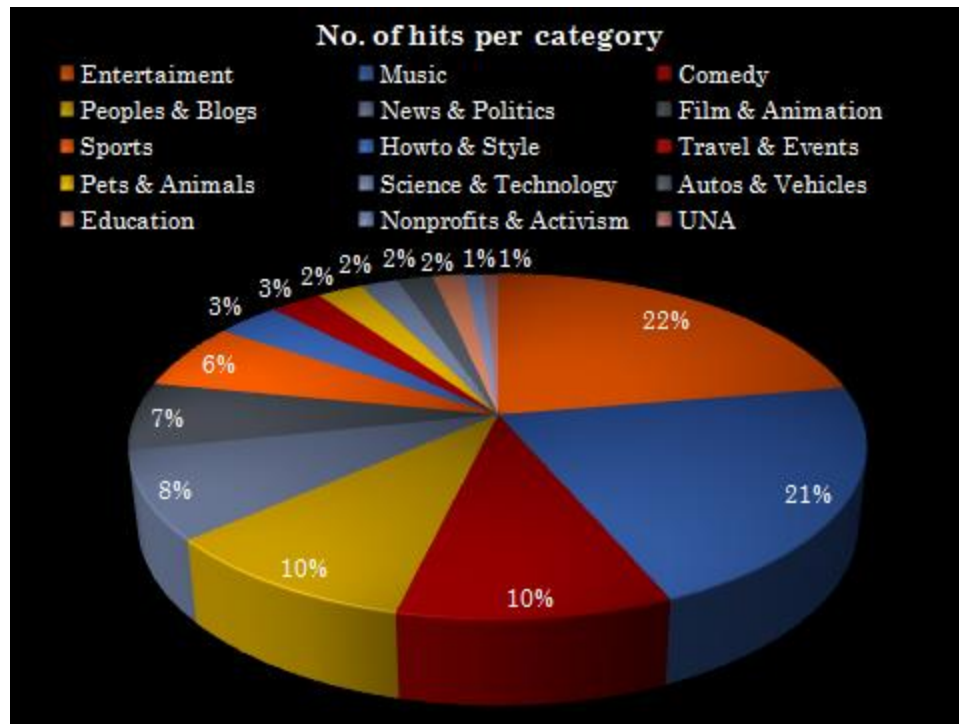
```

[4] KXW16E6, denRA, 711, Entertainment, 306, 6092, 4.12, 25, 31)
[10] LaOpPoYU, jcbce90, 705, People & Blogs, 602, 3888, 4.0, 23, 11)
[4] VYV4Z1zr24, ElmaDODin, 1082, Entertainment, 466, 283, 5.0, 2, 2)
[3] fbfbc9CWRfC, twidesign, 610, Entertainment, 387, 2546, 4.06, 18, 13)
[4] hxyB6D3bvaA, NuoVisoProductions, 1074, Film & Animation, 1197, 175, 4.56, 9, 7)
[8] TbcTcdVq1Q, tksowski, 863, Comedy, 450, 23087, 4.76, 190, 53)
[4] uQq88DIFHC, bmaniac, 597, Film & Animation, 596, 4616, 4.52, 21, 18)
[3] CVFvXmQupg, donkan1je, 555, Entertainment, 226, 2387, 3.94, 16, 3)
[4] w5w5w1k, dwoj0, 650, Film & Animation, 459, 4339, 4.24, 45, 18)
[4] mE9d1vKsE, 711, Entertainment, 587, 2145, 3.77, 13, 6)
[3] CNEdKdCNR8, hmaxx, 795, Film & Animation, 556, 1546, 4.07, 14, 12)
[3] Q6FA9MacYnK, ChrisWorx, 695, Entertainment, 353, 1011, 4.33, 6, 3)
[3] ZCYaw5GyAa, nozzle49, 525, Comedy, 80, 8944331, 4.86, 16551, 6938)
[3] CQzU5TqtW0, mickeymouse, 863, Pets & Animals, 88, 3024767, 4.81, 6968, 4600)
[3] qLXRVHFG-C, seansvoice, 426, Pets & Animals, 309, 3719666, 4.85, 6864, 5428)
[3] CwA8A5 rkZE, greencubix, 338, Pets & Animals, 132, 404628, 4.8, 1358, 209)
[3] MfCzCmKML, dlnherd, 1126, Science & Technology, 209, 3234552, 4.82, 5788, 6093)
[3] W81T1KAW, aureschort, 969, Film & Animation, 234, 162355, 4.62, 468, 238)
[3] N912xpsq7r, lsdnpl735, 503, Music, 335, 724365, 4.83, 1856, 1647)
[3] Lqy59MT03A, joeyfanatic, 558, Music, 254, 446890, 4.86, 921, 529)
[3] BqBvFvPvDm, musicaldopsort, 569, Pets & Animals, 187, 4049478, 4.86, 2955, 945)
[3] T2M6yV6mney, enemigopublico, 764, Music, 316, 1550140, 4.82, 3077, 2609)
[3] KWM037jPwq, NationalGeographic, 909, Pets & Animals, 258, 692556, 4.71, 540, 868)
[3] KLI_h586p9M, timsalmons1982, 851, Music, 306, 3283950, 4.89, 877, 527)
[3] L1L51N38j, uncmick157, 609, Music, 474760, 4.93, 1743, 1267)
[3] KQW0QURB-W, usafar, 461, Pets & Animals, 379, 45076, 4.81, 130, 35)
[3] W5w2Wb, dlpaspe, 769, Music, 137, 224770, 4.9, 639, 378)
[3] CQVQW61Bqk, solasura, 594, Pets & Animals, 659, 174587, 4.72, 1262, 997)
[3] TnaQIrdVd2, raykey2, 618, Entertainment, 192, 610733, 4.73, 1191, 686)
[3] E4VWnt5Usky, bchells, 584, Pets & Animals, 512, 327242, 4.87, 442, 144)
[3] GJ6600-iKpc, rbsza, 468, Music, 343, 309577, 4.79, 564, 381)
[3] crmd_88ERzk, freestyler343, 293, Pets & Animals, 257, 289219, 4.76, 561, 186)
[3] Lzi19vYjplJ, machinima, 1088, Entertainment, 37, 186036, 4.82, 310, 677)
[3] UTYN679BRGq, RockstarGames, 1024, Entertainment, 123, 229527, 4.83, 816, 1408)
[3] e6R8Z0TZXf, machinima, 1098, Entertainment, 39, 53490, 4.82, 117, 254)
[3] K8K6L111B, CRASite, 772, Entertainment, 63, 2708550, 4.76, 3810, 5612)
[3] K5n0L0IAW, machinima, 1090, Entertainment, 36, 248579, 4.66, 524, 683)
[3] EDJ-iWkcJPU, Lassepce, 863, Entertainment, 63, 370075, 4.84, 658, 1284)
[3] M8QK51dosFo, RockstarGames, 772, Entertainment, 63, 437516, 4.74, 1083, 1408)
[3] hKSK1bW9qzw, R2oor, 863, Entertainment, 63, 745466, 4.83, 981, 1955)
[3] 5-re8n0Uqy, Ortmom, 1064, Comedy, 547, 336956, 4.58, 2123, 1384)
[3] 47ENRyJESAM, MgsTheFury404, 1024, Entertainment, 123, 111371, 4.77, 215, 422)
[3] B0uR1JmU2s, RockstarGames, 1016, Entertainment, 48, 108029, 4.8, 254, 402)
[3] S95602a7CfC, matC7B, 1045, Entertainment, 87, 39885, 4.81, 63, 131)
[3] W4545pda, R2OMP, 863, Entertainment, 63, 116939, 4.84, 182, 222)
[3] K0888aR07Y, RockstarGames, 0, UWA, 63, 1939361, 4.79, 1256, 2410)
[3] GSSvZKHuEs, GTASite, 1024, Entertainment, 123, 486752, 4.86, 1098, 1764)
[3] f0cR8qUchQp, scrambledogstv, 1100, Entertainment, 136, 13115, 4.45, 33, 37)
[3] -CYH5Nebec8, rehab077, 771, Entertainment, 186, 103556, 4.3, 277, 220)
grunt>

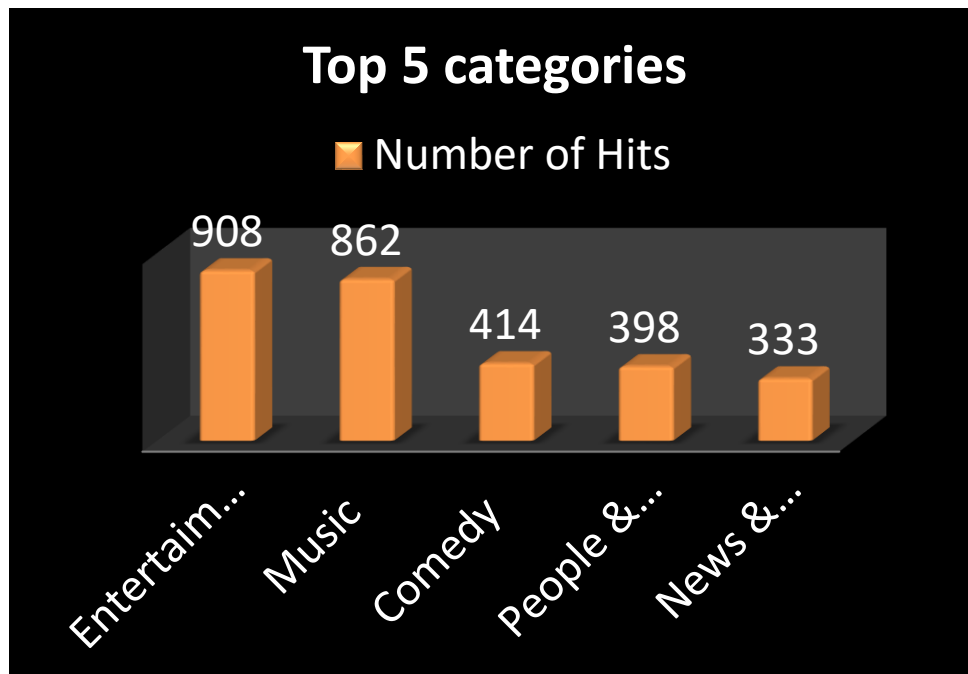
```

Fig 7.2 Data set loaded into pig





Graph 7.1 Count Data visualization



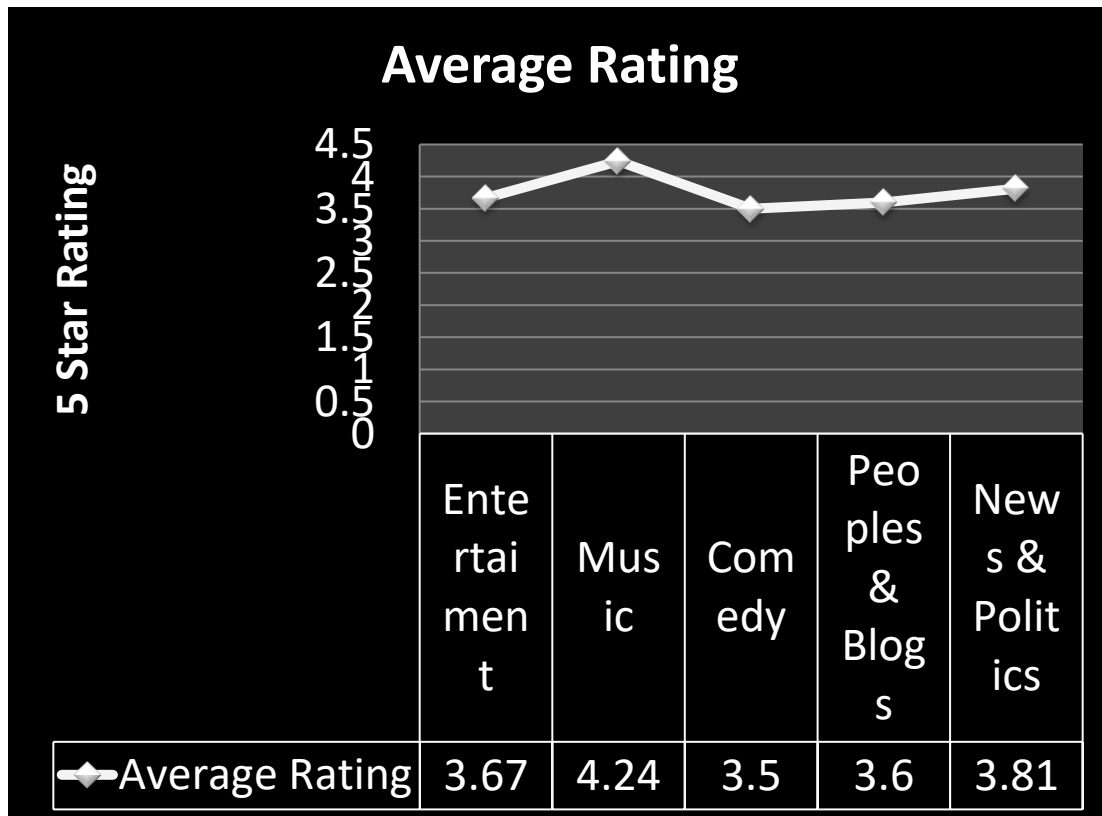
Graph 7.2 Top 5 category no. of hits

```

osboxes@hadoopnode: ~
Success!
Job Stats (time in seconds):
JobID MapTime Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReduceTime Alias Feature Outputs
job_local512994062_0004 1 1 n/a n/a n/a n/a n/a n/a grp.yavg.youtube GROUP_Yr.COMBINED hdfs://localhost:8020/tmp/tmp1092468343/tmp0808305721,
Input(s):
Successfully read 4100 records (54867238 bytes) from: "pig/youtubedata.txt"
Output(s):
Successfully stored 16 records (47112560 bytes) in: "hdfs://localhost:8020/tmp/tmp1092468343/tmp0808305721"
Counters:
Total records written : 16
Total bytes written : 47112560
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_local512994062_0004
2018-08-24 19:31:45,293 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:31:45,294 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:31:45,297 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:31:45,309 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapreducelayer.MapReduceLauncher - Encountered Warning ACCESSING_NON_EXISTENT_FIELD 272 time(s).
2018-08-24 19:31:45,314 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapreducelayer.MapReduceLauncher - Success!
2018-08-24 19:31:45,317 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2018-08-24 19:31:45,331 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2018-08-24 19:31:45,332 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(Music,4.244605567505078)
(Comedy,3.4979710121085676)
(Sports,3.5190909090974135)
(Education,3.341384596910504)
(Entertainment,3.666651979285715)
(Howto & Style,3.3455474463692547)
(People & Blogs,3.594773074210982)
(Pets & Animals,3.3690526284669575)
(News & Politics,3.8094894907495997)
(Travel & Events,3.314274993754266)
(Autos & Vehicles,3.840649334113324)
(Film & Animation,4.087961538479878)
(Science & Technology,3.1832500095234643)
(Nonprofits & Activism,3.2704761823018393)
(.)
grunt>

```

Fig 7.5 Average result



Graph 7.3 Average Rating graph


```

osboxes@hadoopnode: ~
Success!

Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReduceTime Alias Feature Outputs
job_local1214218918_0005 1 1 n/a n/a n/a n/a n/a n/a grp,youtube,ysum GROUP_BY_COMBINER hdfs://localhost:8020/tmp/temp1092448
437/tmp1330338946,

Input(s):
Successfully read 4100 records (68318448 bytes) from: "/pig/youtubedata.txt"

Output(s):
Successfully stored 16 records (58624932 bytes) in: "hdfs://localhost:8020/tmp/temp1092448437/tmp1330338946"

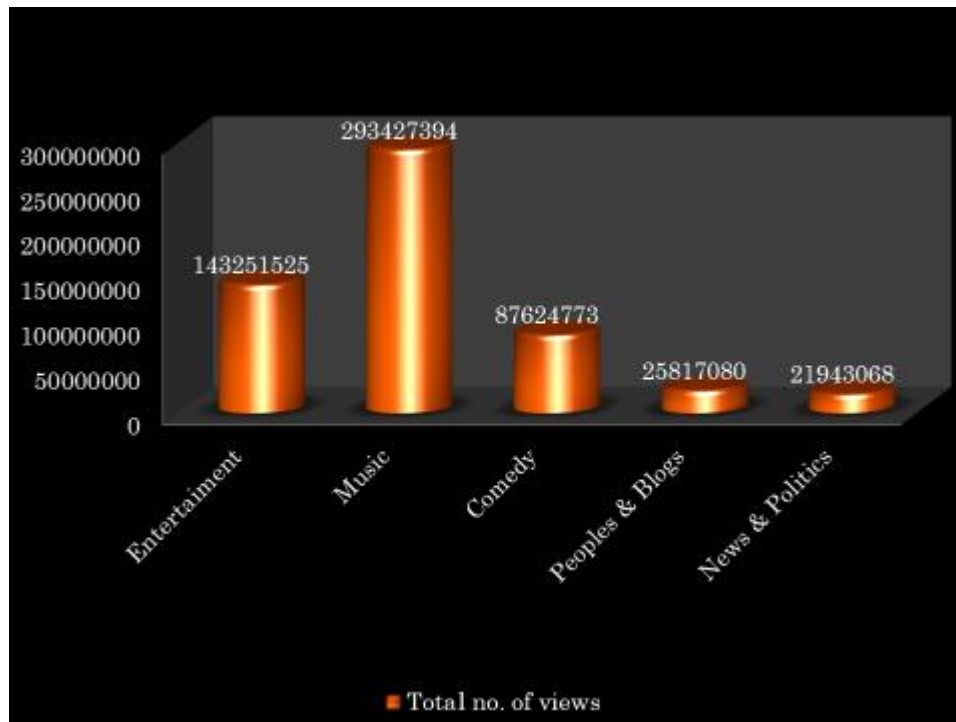
Counters:
Total records written : 16
Total bytes written : 58624932
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1214218918_0005

2018-08-24 19:32:43,927 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:32:43,932 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:32:43,937 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:32:43,947 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning ACCESSING_NON_EXISTENT_FIELD 272 time(s).
2018-08-24 19:32:43,948 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2018-08-24 19:32:43,952 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2018-08-24 19:32:43,964 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2018-08-24 19:32:43,966 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(UNA,9060911)
(Music,293427394)
(Comedy,87624773)
(Sports,25817080)
(Education,1917433)
(Entertainment,143251525)
(Movies & Style,44357536)
(People & Blogs,25817080)
(Pets & Animals,40518898)
(News & Politics,21943068)
(Travel & Events,5105909)
(Autos & Vehicles,3101409)
(Film & Animation,67836805)
(Science & Technology,5598445)
(Nonprofits & Activism,230604)
(, )
grunt>

```

Fig 7.6 Total no. of views result



Graph 7.4 Total no. of views visualization

```
Success!

Job Stats (time in seconds):
JobId  Maps  Reduces  MaxMapTime  MinMapTime  AvgMapTime  MedianMapTime  MaxReduceTime  MinReduceTime  AvgReduceTime  MedianReduceTime  Alias  Feature Outputs
job_local1593420661_0004  1  1  n/a  n/a  n/a  n/a  n/a  n/a  gip.youtube,yum  GROUP_BY,COMBINEA  hdfs://localhost:8020/tmp/tem
p1092468336/tmp66068336,

Input(s):
Successfully read 4109 records (81769538 bytes) from: "p/g/youtubedata.txt"

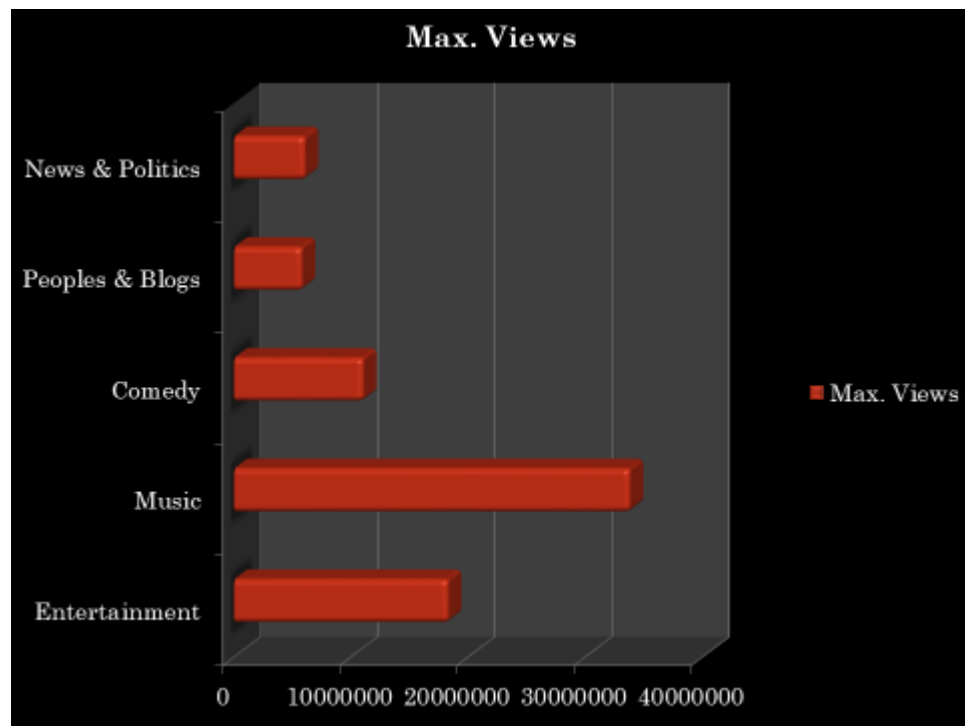
Output(s):
Successfully stored 16 records (70137244 bytes) in: "hdfs://localhost:8020/tmp/1092468336/tmp66068336"

Counters:
Total records written : 16
Total bytes written : 70137244
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1593420661_0004

2018-08-24 19:33:57,436 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:33:57,450 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:33:57,458 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:33:57,478 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning ACCESSING_NON_EXISTENT_FIELD 272 time(s).
2018-08-24 19:33:57,482 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2018-08-24 19:33:57,496 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2018-08-24 19:33:57,713 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2018-08-24 19:33:57,716 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(UMA,5060911)
(Music,293427394)
(Comedy,87624772)
(Sports,25838327)
(Education,1917433)
(Entertainment,143251325)
(Howto & Style,24357349)
(People & Blogs,25817080)
(Pets & Animals,60518898)
(News & Politics,21943068)
(Travel & Events,5105907)
(Autos & Vehicles,3101409)
(Film & Animation,87836805)
(Science & Technology,558846)
(Nonprofits & Activism,338604)
(,.)
grunt>
```

Fig 7.7 Max. no. of views result



Graph 7.5 Max views visualization

```

osboxes@hadoopnode: ~
Success!

Job Stats (time in seconds):
JobId  Maps  Reduces  MaxMapTime  MinMapTime  AvgMapTime  MedianMapTime  MaxReduceTime  MinReduceTime  AvgReduceTime  MedianReduceTime  Alias  Feature Outputs
job_local273682735_0007  1  n/a  n/a  n/a  n/a  n/a  n/a  n/a  grp,yavgl,youtube  GROUP_BY_COMBINEA  hdf://localhost:8020/tmp/temp109244843/tmp-651510005,

Input(s):
Successfully read 4100 records (95220628 bytes) from: "/pig/youtubedata.txt"

Output(s):
Successfully stored 16 records (81649616 bytes) in: "hdfs://localhost:8020/tmp/temp109244843/tmp-651510005"

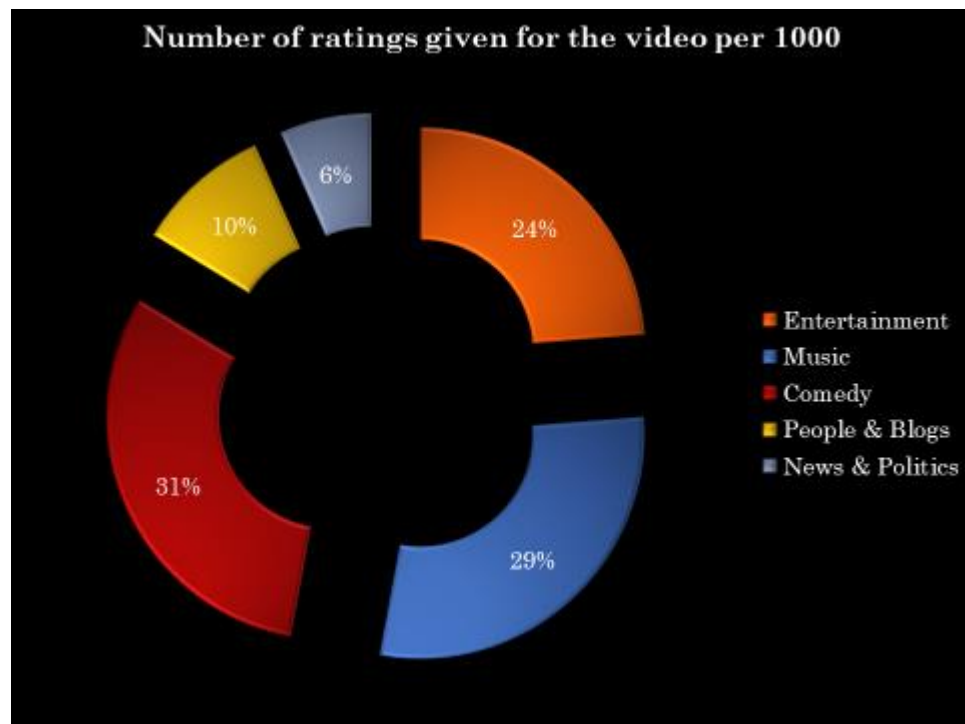
Counters:
Total records written : 16
Total bytes written : 81649616
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local273682735_0007

2018-08-24 19:36:01,581 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:36:01,588 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:36:01,590 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2018-08-24 19:36:01,619 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning ACCESSING_NON_EXISTING_FIELD 272 time(s).
2018-08-24 19:36:01,620 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2018-08-24 19:36:01,623 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2018-08-24 19:36:01,635 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2018-08-24 19:36:01,636 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer - Total input paths to process : 1
( UNK , 273.25)
(Music, 891.6245501160093)
(Comedy, 931.5048309179744)
(Sports, 170.01195211232909)
(Education, 57.69230769230769)
(Entertainment, 718.5055066079295)
(Music & Style, 415.7153294671533)
(People & Blogs, 293.8945728462216)
(Pets & Animals, 1721.221052631579)
(News & Politics, 199.75075075075074)
(Travel & Events, 397.11607142857144)
(Autos & Vehicles, 30.1449831949832)
(Film & Animation, 288.24615384615385)
(Science & Technology, 160.4625)
(Nonprofits & Activism, 16.88095238095238)
()
grant>

```

Fig 7.8 Ratings per 1000 video result



Graph 7.6 Rating per 100 video visualization

CHAPTER 8

CONCLUSION

The task of big data analysis is not only important but also a necessity. In fact many organizations that have implemented Big Data are realizing significant competitive advantage compared to other organizations with no Big Data efforts. The project is intended to analyze the YouTube Big Data and come up with significant insights which cannot be determined otherwise.

The output results of YouTube data analysis project show key insights that can be extrapolated to other use cases as well. One of the output results describes that for a specific video id, how many likes were received. The number of likes -- or "thumbs-up" - a video had has a direct significance to the YouTube video's ranking, according to YouTube Analytics. So if a company posts its video on YouTube, then the number of YouTube likes the company has could determine whether the company or its competitors appear more prominently in YouTube search results.

Another output result gives us insights on if there is a pattern of affinity of interests for certain video category. This can be done by analyzing the comments count. For e.g., if the company falls under 'Comedy' or 'Education' category, a meaningful discussion in the form of comments can be triggered on YouTube. A comment analysis can further be conducted to understand the attitude of people towards the specific video.

CHAPTER 9

FUTURE ENHANCEMENTS

The future work would include extending the analysis of YouTube data using other Big Data analysis Technologies like Pig and MapReduce and do a feature comparison analysis. It would be interesting to see which technology fares better as compared to the other ones. One feature that is not added in the project is 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. However, if the output is too large and complex, the output can be interfaced in a GUI format to display the results. The data can then be presented in different format including pie-charts and graphs for better user experience.

Another possible extension of this project could be the YouTube Comment Analysis project. The current scope of the project includes analyzing the statistics for a channel/category including view counts, likes, dislikes, country wise view etc. By identifying classifying/categorizing the polarity of the words, sentiment analysis or opinion mining can be performed for a specific video. This would tell us writer's attitude towards a particular product or a given subject. Using Sentiment Analysis, we can determine if the general attitude of people is positive, negative or neutral towards a specific subject/video.

CHAPTER 10

REFERENCES

10.1 URLs

- [1] <http://hadoop.apache.org/>
- [2] www.w3schools.com
- [3] www.google.com
- [4] www.tutorialspoint.com

10.2 BOOKS

- [1] Hadoop in action
- [2] Hadoop Operations
- [3] Big Data Black Book