TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

VIỆN TRÍ TUỆ NHÂN TẠO

-----*-----**

BÁO CÁO MÔN HỌC KỸ THUẬT VÀ CÔNG NGHỆ DỮ LIỆU LỚN

ĐỀ TÀI

PHÂN TÍCH DỮ LIỆU YOUTUBE

Nhóm sinh viên thực hiện:

1. Lê Nguyên Vũ - 22022544
2. Nguyễn Trọng Khánh - 22022603
3. Chu Thân Nhất - 22022578

Giảng viên hướng dẫn: TS. Trần Hồng Việt

ThS.Ngô Minh Hương

HÀ NỘI, 5/2024

MỞ ĐẦU

Trong thời đại công nghệ số, Big Data đang trở thành một trong những yếu tố then chốt giúp thúc đẩy sự phát triển của nhiều lĩnh vực như y tế, giáo dục, giao thông, và thương mại điện tử. Từ tháng 8/2015, Big Data đã thoát khỏi bảng xếp hạng Cycle Hype của Gartner và được công nhận như một công nghệ chủ đạo, tạo ra những bước đột phá mạnh mẽ trong việc khai thác dữ liệu. Dữ liệu lớn chứa đựng một lượng thông tin vô giá, và việc trích xuất thành công những thông tin này có thể mang lại những lợi ích vượt trội cho các tổ chức và doanh nghiệp.

Hadoop, một framework hàng đầu trong việc xử lý và phân tích dữ liệu lớn, đã khẳng định vị trí cốt lõi của mình trong việc lưu trữ và xử lý khối lượng lớn dữ liệu bằng cách áp dụng mô hình lập trình MapReduce. Hadoop không chỉ giúp quản lý dữ liệu hiệu quả mà còn mở rộng khả năng phân tích, đưa ra những hiểu biết sâu sắc từ dữ liệu.

Từ thực tế này, chúng em đã chọn đề tài: **"Phân tích dữ liệu YouTube sử dụng Hadoop"** để làm bài báo cáo kết thúc môn học của mình. Trong bài báo cáo, chúng em đã ứng dụng các tính năng của Hadoop để phân tích các thuộc tính như **lượt xem, bình luận của video,** qua đó khám phá những yếu tố ảnh hưởng đến sự phổ biến của nội dung trên nền tảng YouTube.

Báo cáo được trình bày gồm 4 chương:

- **Chương 1:** Tổng quan về dữ liệu lớn và Hadoop.
- **Chương 2:** Tìm hiểu về cấu trúc dữ liệu YouTube.
- **Chương 3:** Phân tích dữ liệu YouTube sử dụng Hadoop.
- **Chương 4:** Kết luận và hướng phát triển.

MỤC LỤC

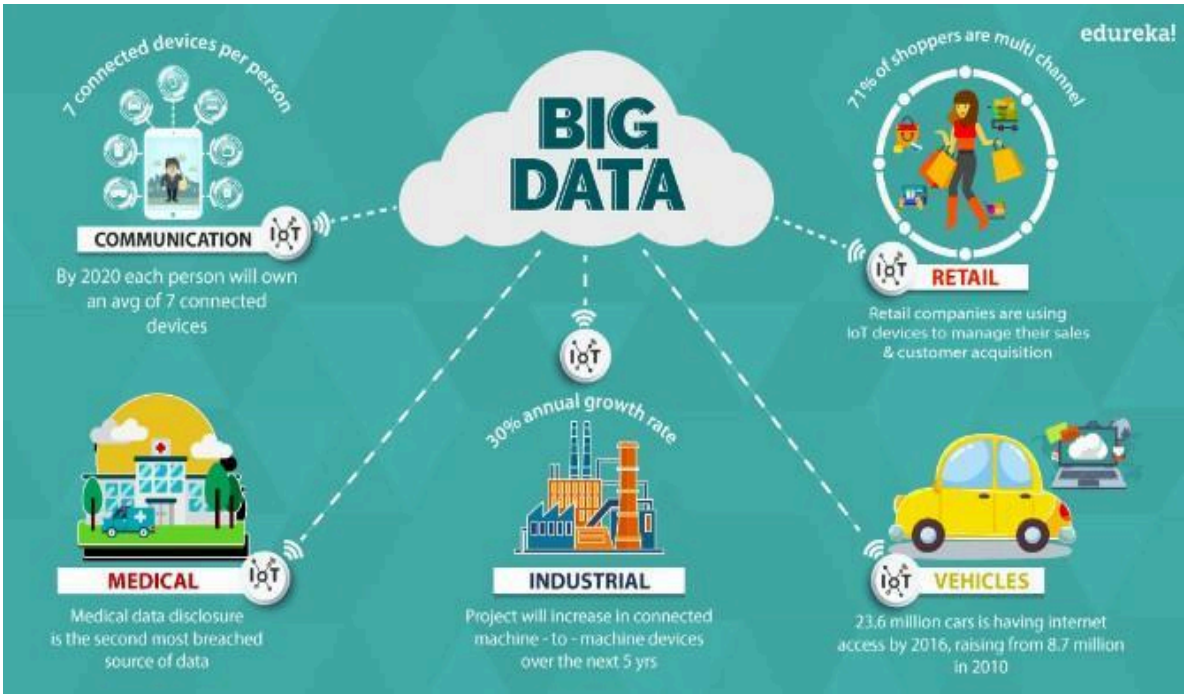
CHƯƠNG 1: TỔNG QUAN VỀ DỮ LIỆU LỚN

1.1 Định nghĩa.

Theo wikipedia: Dữ liệu lớn là một thuật ngữ chỉ bộ dữ liệu lớn hoặc phức tạp mà các phương pháp truyền thống không đủ các ứng dụng để xử lý dữ liệu này.

Theo Gartner : Dữ liệu lớn là những nguồn thông tin có đặc điểm chung khối lượng lớn, tốc độ nhanh và dữ liệu định dạng dưới nhiều hình thức khác nhau,do đó muốn khai thác được phải đòi hỏi phải có hình thức mới để đưa ra quyết định khám phá và tối ưu hóa quy trình.

Dữ liệu đến từ rất nhiều nguồn khác nhau:



Hình 1. Minh họa nguồn gốc của dữ liệu.

Một số lợi ích có thể mang lại như: Cắt giảm chi phí, tiết kiệm thời gian và giúp tối ưu hóa sản phẩm, hỗ trợ con người đưa ra những quyết định dúng và hợp lý hơn.

1.2 Đặc trưng cơ bản của dữ liệu lớn.

(1) Khối lượng lớn (Volume): Khối lượng dữ liệu rất lớn và đang ngày càng tăng lên, tính đến 2014 thì có thể trong khoảng vài trăm terabyte.

(2) Tốc độ (Velocity): Khối lượng dữ liệu gia tăng rất nhanh.

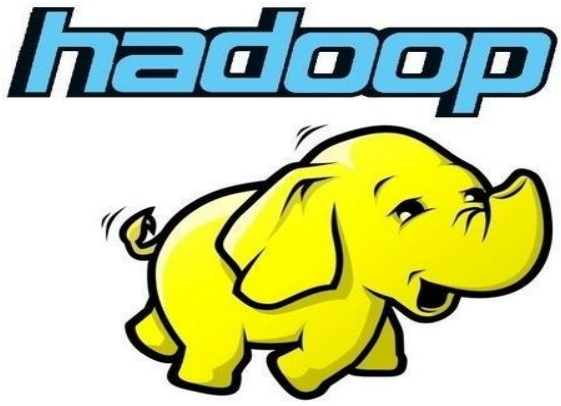
(3) Đa dạng (Variety): Ngày nay hơn 80% dữ liệu được sinh ra là phi cấu trúc(tài liệu, blog, hình ảnh,...)

(4) Độ tin cậy/chính xác(Veracity): Bài toán phân tích và loại bỏ dữ liệu thiếu chính xác và nhiễu đang là tính chất quan trọng của bigdata.

(5) Giá trị(Value): Giá trị thông tin mang lại.

1.3 Tổng quan về Hadoop.

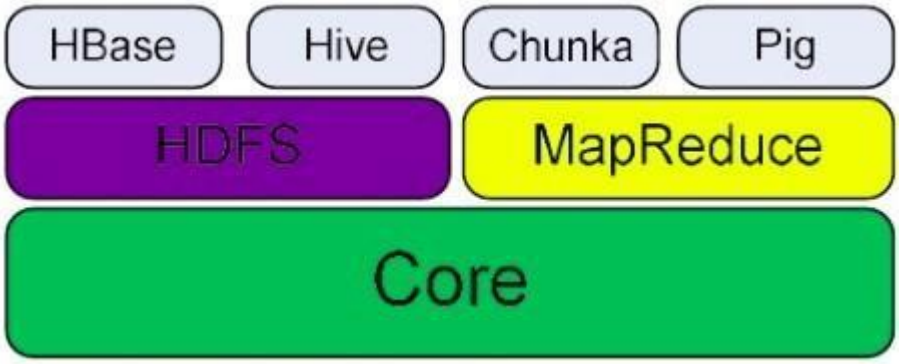
Theo apache hadoop: Apache Hadoop là một framework dùng để chạy những ứng dụng trên 1 cluster lớn được xây dựng trên những phần cứng thông thường.



Hình 2.Biểu tượng của Hadoop

+ Các thành phần của hadoop: Core, MapReduce engine, HDFS, HBase, Hive, Pig, Chukwa,..

Tuy nhiên *tập chung* vào 2 thành phần quan trọng nhất: HDFS và MapReduce.



Hình 3. Thành phần của Hadoop

+ Hadoop hiện thực mô hình Mapreduce, đây là mô hình mà ứng dụng sẽ được chia nhỏ ra thành nhiều phân đoạn khác nhau, và các phần này sẽ được chạy song song trên nhiều node khác nhau.

+ Thêm vào đó, Hadoop cung cấp 1 hệ thống file phân tán (HDFS) cho phép lưu trữ dữ liệu lên trên nhiều node. Cả Mapreduce và HDFS đều được thiết kế sao cho framework sẽ tự động quản lý được các lỗi, các hư hỏng về phần cứng của các node.

=> *Kết luận:* Là một framework cho phép phát triển các ứng dụng phân tán. Viết bằng java.

1.4 *Tổng quan về MapReduce.*

➢ *Định nghĩa:* Theo Google, MapReduce là mô hình dùng cho xử lý tính toán song song và phân tán trên hệ thống phân tán.

B1: Phân rã từ nghiệp vụ chính (do người dùng muốn thể hiện) thành các công việc con để chia từng công việc con này về các máy tính trong hệ thống thực hiện xử lý một cách song song.

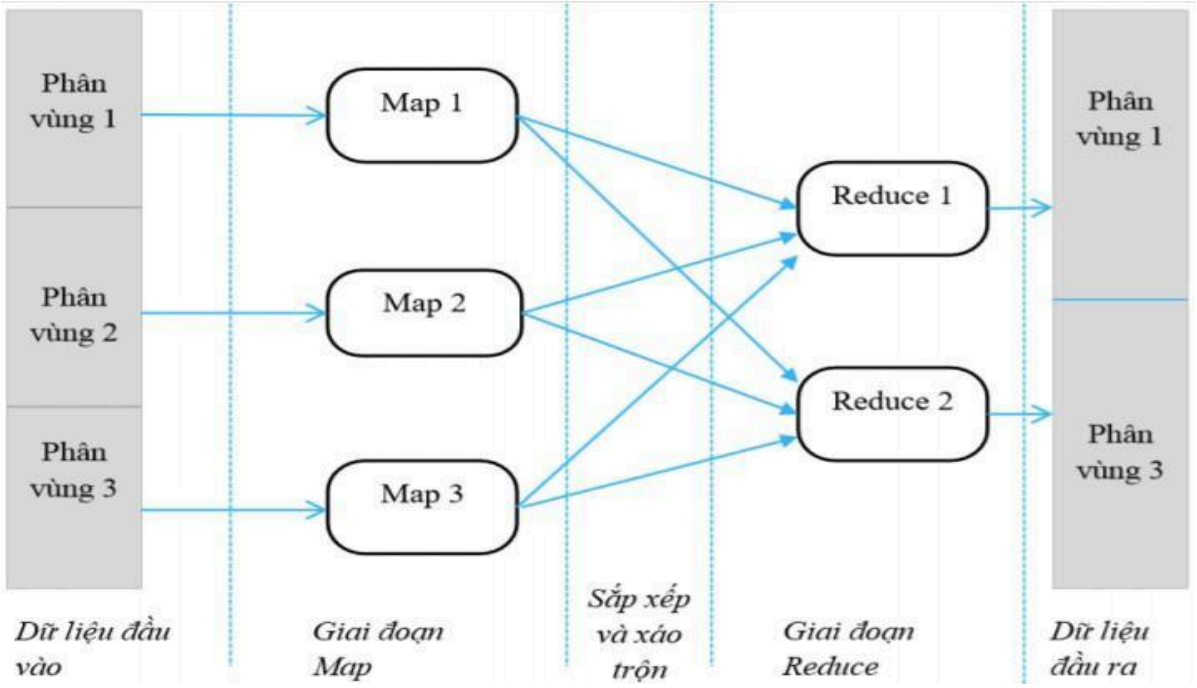
B2: Thu thập lại các kết quả.

➢ Ứng dụng của MapReduce:

+ Dữ liệu cần xử lý kích thước lớn.

+ Các ứng dụng thực hiện xử lý, phân tích dữ liệu, thời gian xử lý đáng kể, có thể tính bằng phút, giờ, …

➢ Thực thi mô hình MapReduce:



Hình 4. Thực thi mô hình Mapreduce

+ Hàm Map : Hàm Map tiếp nhận mảnh dữ liệu input, rút trích thông tin cần thiết các từng phần tử (ví dụ: lọc dữ liệu, hoặc trích dữ liệu) tạo kết quả trung gian

+ Hàm Reduce: tổng hợp kết quả trung gian, tính toán để cho kết quả cuối cùng.

CHƯƠNG 2: CẤU TRÚC DỮ LIỆU YOUTUBE

2.1 Tổng quan dữ liệu Youtube

```
RangeIndex: 40949 entries, 0 to 40948
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   video_id                40949 non-null  object
 1   trending_date           40949 non-null  object
 2   title                   40949 non-null  object
 3   channel_title           40949 non-null  object
 4   category                40949 non-null  object
 5   publish_time            40949 non-null  object
 6   tags                    40949 non-null  object
 7   views                   40949 non-null  int64
 8   likes                   40949 non-null  int64
 9   dislikes                40949 non-null  int64
 10  comment_count           40949 non-null  int64
 11  thumbnail_link          40949 non-null  object
 12  comments_disabled       40949 non-null  bool
 13  ratings_disabled        40949 non-null  bool
 14  video_error_or_removed  40949 non-null  bool
 15  description             40379 non-null  object
```

Bảng dữ liệu YouTube trên cung cấp một cái nhìn tổng quan về các video trên nền tảng YouTube, với tổng cộng 40.949 bản ghi. Dữ liệu này chứa các thông tin quan trọng về các video, bao gồm các đặc điểm cơ bản của video, mức độ tương tác của người xem, trạng thái của video, và các yếu tố hình ảnh liên quan.

Cụ thể, các cột trong dữ liệu có thể được chia thành các nhóm chính:

1. **Thông tin cơ bản về video**: Bao gồm video_id , title , channel_title , category , publish_time , và description . Các thông tin này giúp xác định đặc điểm của mỗi video và phân loại video theo các tiêu chí như thể loại hoặc thời gian phát hành.
2. **Mức độ tương tác với video**: Các cột như views , likes , dislikes , và comment_count phản ánh mức độ phổ biến và sự tương tác của người xem với video.
3. **Trạng thái của video**: Các cột comments_disabled, ratings_disabled, và video_error_or_removed cho thấy trạng thái hiện tại của video trên nền tảng, cho phép xác định các video không còn hoạt động hoặc bị hạn chế tương tác.
4. **Thông tin về hình ảnh**: Cột thumbnail_link chứa liên kết đến hình ảnh thu nhỏ của video, đóng vai trò quan trọng trong việc thu hút sự chú ý của người xem.

Dữ liệu này cung cấp một nền tảng vững chắc để phân tích và đánh giá sự phổ biến của video

trên YouTube, các yếu tố ảnh hưởng đến mức độ tương tác, và các xu hướng nội dung nổi bật trên nền tảng. Các phân tích chi tiết về ý nghĩa của từng thuộc tính sẽ được trình bày ở các phần sau của báo cáo.

2.2 Ý nghĩa của các thuộc tính trong tập dữ liệu USvideos0.csv.

Bộ dữ liệu YouTube chứa các thuộc tính phản ánh chi tiết về video, mức độ tương tác của người xem và trạng thái video. Dưới đây là ý nghĩa cụ thể của từng thuộc tính:

1. video_id : Đây là mã định danh duy nhất cho mỗi video trên YouTube, giúp phân biệt và truy xuất video dễ dàng.
2. trending_date : Ghi nhận ngày mà video bắt đầu xuất hiện trong danh sách thịnh hành, cho phép phân tích xu hướng video qua thời gian.
3. title (Tiêu Đề Video): Phần văn bản mô tả ngắn gọn nội dung video, là yếu tố quan trọng để thu hút người xem và tối ưu hóa tìm kiếm.
4. channel_title : Tên kênh YouTube đăng tải video, phản ánh nguồn sáng tạo nội dung và mức độ phổ biến của các kênh.
5. category : Phân loại video thành các nhóm nội dung như giải trí, giáo dục, âm nhạc, tin tức, v.v., giúp phân tích sự phân bố và xu hướng nội dung.
6. publish_time : Thời điểm video được đăng tải, hữu ích để phân tích mối quan hệ giữa thời gian đăng và sự thành công của video.
7. tags : Các từ khóa mô tả nội dung video, giúp cải thiện khả năng tìm kiếm và phản ánh chủ đề chính của video.
8. views : Số lượng người đã xem video, là chỉ số đánh giá mức độ phổ biến và sức hút của video.
9. likes : Số lượt người dùng nhấn nút thích để thể hiện sự yêu thích nội dung video.
10. dislikes : Số lượt người dùng nhấn nút không thích, phản ánh sự không hài lòng hoặc không đồng tình với nội dung.
11. comment_count : Tổng số bình luận của video, thể hiện mức độ tương tác và thảo luận của người xem.
12. thumbnail_link : Liên kết đến hình ảnh đại diện của video, đóng vai trò quan trọng trong việc thu hút lượt nhấp chuột.
13. comments_disabled : Cho biết tính năng bình luận có bị tắt hay không, thường liên quan đến kiểm duyệt hoặc quyết định của người sáng tạo nội dung.
14. ratings_disabled : Cho biết tính năng xếp hạng (thích/không thích) có bị tắt hay không, giúp phân tích cách người sáng tạo quản lý phản hồi từ khán giả.
15. video_error_or_removed : Thể hiện trạng thái video đã bị lỗi hoặc bị xóa, thường do vi phạm chính sách hoặc sự cố kỹ thuật.
16. description : Phần mô tả chi tiết về nội dung video, hỗ trợ tìm kiếm và cung cấp thêm thông tin cho người xem.

Các thuộc tính này là cơ sở quan trọng để phân tích sự thành công của video, xu hướng nội dung, và mức độ tương tác của khán giả trên nền tảng YouTube.

2.3 Các thuộc tính dữ liệu cần phân tích.

```
levutb@levutb-VirtualBox:~$ hdfs dfs -cat /YoutubeOutput/*
Action/Adventure        2716784
Autos & Vehicles        45776612
Classics        5511207
Comedy  69420250
Documentary     501447435
Drama   6600369
Family  566156899
Foreign 232427080
Horror  14399111
Movies  55184947
Music   7119588
Sci-Fi/Fantasy  14491116
Science & Technology    2337041
Shorts  298489
Thriller        262619232
Action/Adventure        98094297
Autos & Vehicles        107272677
Classics        10765560
Comedy  168569794
Documentary     64600253
Drama   116613196
Family  451004621
Foreign 41602888
Horror  76339352
Movies  77601109
Music   104945036
Sci-Fi/Fantasy  21202882
Science & Technology    32540510
Shorts  95469
Technology and Sports Entertainment     287742
Thriller        40580404
Action/Adventure        19385730
Autos & Vehicles        128939027
Classics        9240939
Comedy  315007395
Documentary     145819446
```

Trong bài toán này chúng ta sử dụng ba thuộc tính để phân tích data bao gồm: category, views và comment_count.

Dựa vào các thuộc tính trên ta tính được số lượng views cho từng thể loại như hình ảnh minh họa bên trên.

Dựa vào các số liệu được tổng hợp như trên dễ dàng có thể thấy được mức độ hứng thú của người xem qua từng thể loại:

- Action/Adventure : 271674
- Auto & Vehicles : 45776612
- Music : 7119588

…..

Qua các thống kê ở trên cho ta khái quát thị hiếu của người xem dựa trên các thể loại trending. Từ đó, người dùng có thể định hướng mục tiêu, định hướng của nội dung kênh khi bắt đầu tham gia vào Youtube.

CHƯƠNG 3: PHÂN TÍCH DỮ LIỆU YOUTUBE SỬ DỤNG HADOOP

3.1 Ý tưởng phân tích dữ liệu

- Map: Đọc vào dữ liệu.
- Partitioner: Chia dữ liệu thành nhiều phần nhỏ.
- Reduce: Tập hợp dữ liệu và đưa ra kết quả

3.2 Mapper, Reducer và Partitioner

- Mapper: trích xuất dữ liệu từ 3 cột category, views, comment_count và tổng hợp thành các cặp (category, (views, comment_count))

```java
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

import java.io.IOException;

public class PopularBasedOnViewsMapper extends Mapper<Object, Text, Text, Text> {
    @Override
    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
        String line = value.toString();
        String[] fields = line.split(",");

        try {
            // Extract fields (adjust indexes based on dataset structure)
            String categoryId = fields[4].trim();        // category_id
            String views = fields[7].trim();            // views
            String commentCount = fields[10].trim();    // comment_count

            // Emit category_id as key and "views,comment_count" as value
            context.write(new Text(categoryId), new Text(views + "," + commentCount));
        } catch (Exception e) {
            // Skip invalid rows
            System.err.println("Invalid input: " + line);
        }
    }
}
```

- Partitioner: chia job phải làm thành 6 tasks dựa vào comment_count. category dưới 100 comments được xếp vào task 0, từ 100 đến 500 comments vào task 1, 500 đến 1000 comments vào task 2, 1000 đến 5000 comments vào task 3, 5000 đến 10000 comments vào task 4, trên 10000 comments vào task 5

```java
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Partitioner;

public class PopularBasedOnViewsPartitioner extends Partitioner<Text, Text> {
    @Override
    public int getPartition(Text key, Text value, int numReduceTasks) {
        try {
            // Extract comment_count from the value
            String[] fields = value.toString().split(",");
            int commentCount = Integer.parseInt(fields[1].trim()); // comment_count is the second field in value

            // Partition based on comment_count ranges
            if (commentCount < 100) {
                return 0;
            } else if (commentCount < 500) {
                return 1;
            } else if (commentCount < 1000) {
                return 2;
            } else if (commentCount < 5000) {
                return 3;
            } else if (commentCount < 10000) {
                return 4;
            } else {
                return 5;
            }
        } catch (NumberFormatException e) {
            // Default to partition 0 for invalid data
            return 0;
        }
    }
}
```

- Reducer: đưa ra cặp với key là category và value là tổng số views trong partition.9

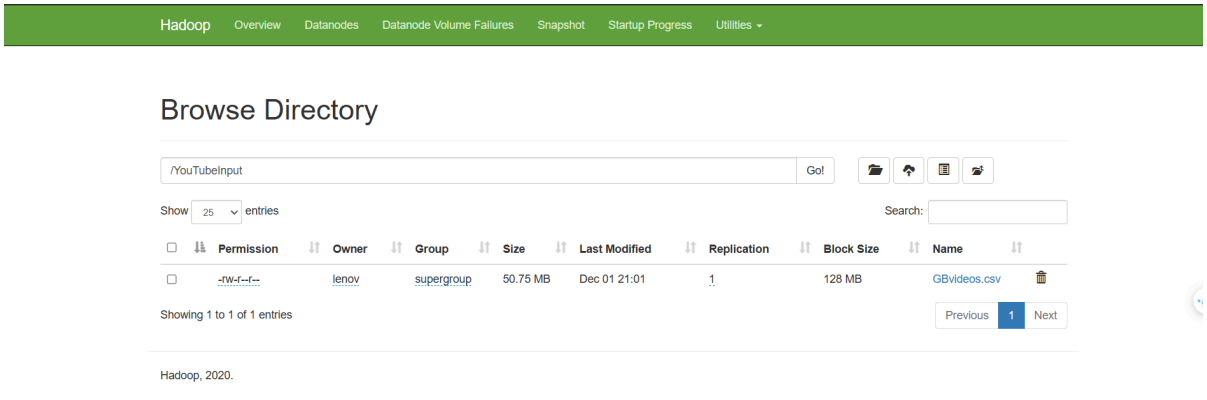
```java
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;

public class PopularBasedOnViewsReducer extends Reducer<Text, Text, Text, Text> {
        @Override
        public void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
            long totalViews = 0;

            for (Text value : values) {
                String[] fields = value.toString().split(",");
                try {
                    // Accumulate views
                    totalViews += Long.parseLong(fields[0].trim()); // views is the first field in value
                } catch (NumberFormatException e) {
                    System.err.println("Invalid views: " + value.toString());
                }
            }

            // Skip categories with 0 views
            if (totalViews > 0) {
                context.write(key, new Text(String.valueOf(totalViews)));
            }
        }
}
```

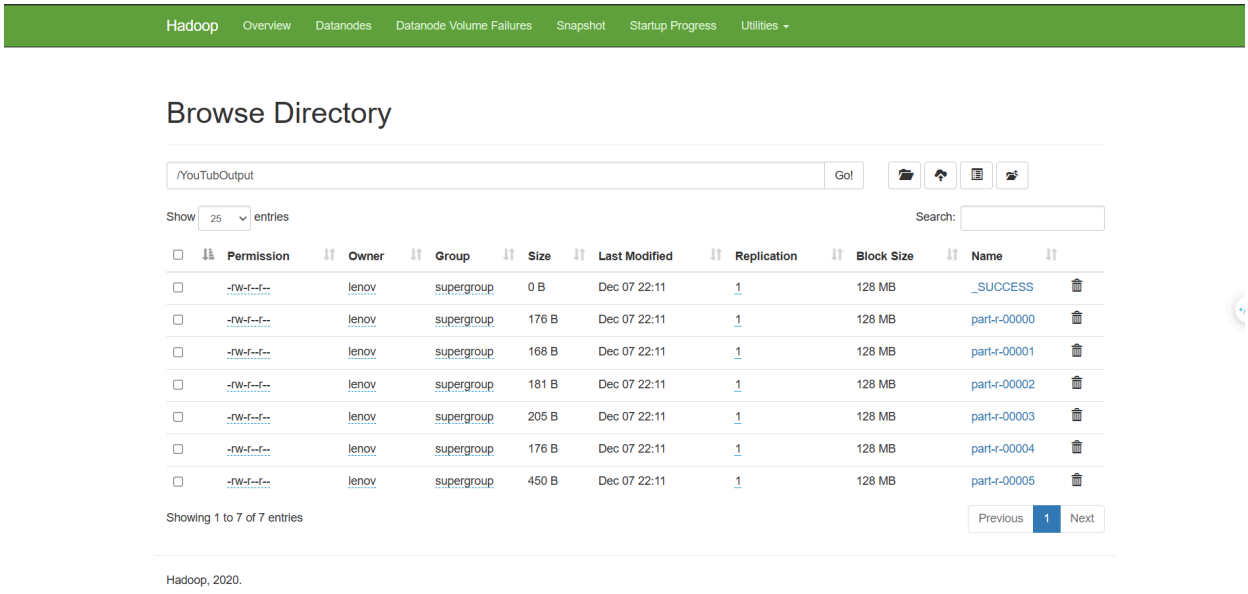
- Driver: đưa cho job các class và đưa ra kết quả ở output file

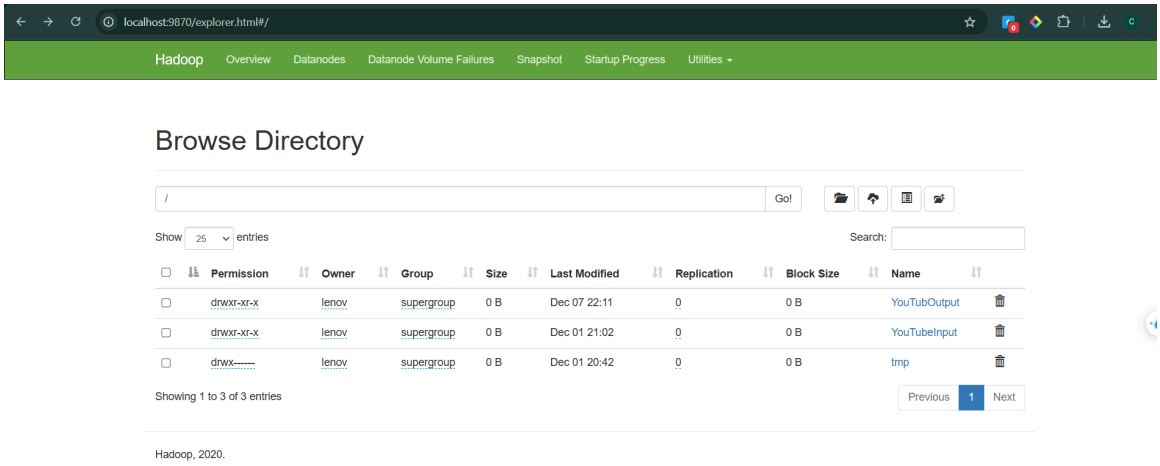
```java
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class PopularVideosJob {
    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Usage: PopularCategoryDriver <input path> <output path>");
            System.exit(-1);
        }

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Popular Category Based on Views");

        job.setJarByClass(PopularVideosJob.class);
        job.setMapperClass(PopularBasedOnViewsMapper.class);
        job.setPartitionerClass(PopularBasedOnViewsPartitioner.class);
        job.setReducerClass(PopularBasedOnViewsReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        // Set number of reducers to match partitions
        job.setNumReduceTasks(6);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

3.3 Demo kết quả.

1. Kết quả chạy thành công

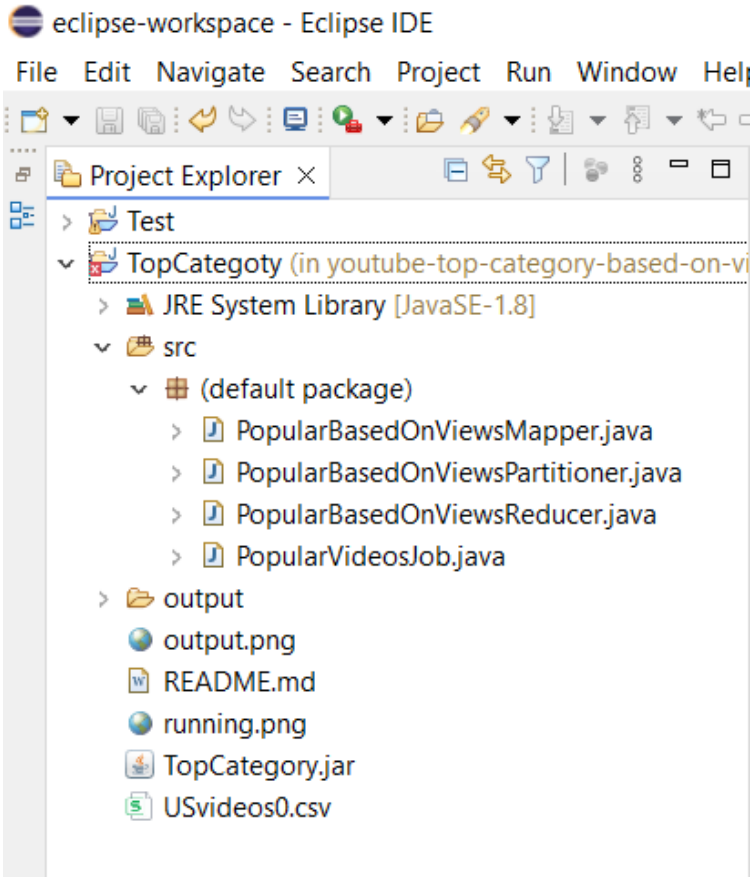

Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/YouTubeInput
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
-rw-r--r--
lenov
supergroup
50.75 MB
Dec 01 21:01
1
128 MB
GBvideos.csv
Showing 1 to 1 of 1 entries
Previous
1
Next
Hadoop, 2020.


Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/YouTubOutput
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
-rw-r--r--
lenov
supergroup
0 B
Dec 07 22:11
1
128 MB
_SUCCESS
-rw-r--r--
lenov
supergroup
176 B
Dec 07 22:11
1
128 MB
part-r-00000
-rw-r--r--
lenov
supergroup
168 B
Dec 07 22:11
1
128 MB
part-r-00001
-rw-r--r--
lenov
supergroup
181 B
Dec 07 22:11
1
128 MB
part-r-00002
-rw-r--r--
lenov
supergroup
205 B
Dec 07 22:11
1
128 MB
part-r-00003
-rw-r--r--
lenov
supergroup
176 B
Dec 07 22:11
1
128 MB
part-r-00004
-rw-r--r--
lenov
supergroup
450 B
Dec 07 22:11
1
128 MB
part-r-00005
Showing 1 to 7 of 7 entries
Previous
1
Next
Hadoop, 2020.



2. Demo chạy chương trình

- Cấu trúc của thư mục project



- Chạy chương trình
  - Tạo thư mục YouTubeInput

```
hdfs dfs -mkdir /YouTubeInput
```

  - Đẩy dữ liệu USvideo0.csv

```
hdfs dfs -put /Downloads/youtube-top-category-based-on-views/USvideos0.csv /YouTubeInput
```

  - Tạo và thực thi tệp jar và lưu kết quả vào thư mục output trong hdfs:

```
hadoop jar /home/hadoop/TopCategory.jar PopularVideosJob /YouTubeInput /YouTubeOutput
```

  - Xem kết quả:

```
hdfs dfs -cat /YouTubeOutput/*
```

- Kết quả

☐	-rw-r--r--	lenov	supergroup	0 B	Dec 07 22:11	1	128 MB	_SUCCESS
☐	-rw-r--r--	lenov	supergroup	176 B	Dec 07 22:11	1	128 MB	part-r-00000
☐	-rw-r--r--	lenov	supergroup	168 B	Dec 07 22:11	1	128 MB	part-r-00001
☐	-rw-r--r--	lenov	supergroup	181 B	Dec 07 22:11	1	128 MB	part-r-00002
☐	-rw-r--r--	lenov	supergroup	205 B	Dec 07 22:11	1	128 MB	part-r-00003
☐	-rw-r--r--	lenov	supergroup	176 B	Dec 07 22:11	1	128 MB	part-r-00004
☐	-rw-r--r--	lenov	supergroup	450 B	Dec 07 22:11	1	128 MB	part-r-00005

```
levutb@levutb-VirtualBox:~$ hdfs dfs -cat /YoutubeOutput/*
Action/Adventure        2716784
Autos & Vehicles        45776612
Classics        5511207
Comedy  69420250
Documentary     501447435
Drama   6600369
Family  566156899
Foreign 232427080
Horror  14399111
Movies  55184947
Music   7119588
Sci-Fi/Fantasy  14491116
Science & Technology    2337041
Shorts  298489
Thriller        262619232
Action/Adventure        98094297
Autos & Vehicles        107272677
Classics        10765560
Comedy  168569794
Documentary     64600253
Drama   116613196
Family  451004621
Foreign 41602888
Horror  76339352
Movies  77601109
Music   104945036
Sci-Fi/Fantasy  21202882
Science & Technology    32540510
Shorts  95469
Technology and Sports Entertainment     287742
Thriller        40580404
Action/Adventure        19385730
Autos & Vehicles        128939027
Classics        9240939
Comedy  315007395
Documentary     145819446
```

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận.

Big data mang đến cơ hội và thách thức lớn, đặc biệt trong việc khai thác giá trị từ khối dữ liệu khổng lồ. Hadoop, với mô hình MapReduce, là công cụ hiệu quả giúp chia nhỏ công việc xử lý, phân tán trên các nút tính toán và tập hợp kết quả. Trong đề tài này, nhóm đã áp dụng Hadoop để phân tích dữ liệu YouTube từ tập USvideos.csv. Đề tài "Phân tích dữ liệu YouTube sử dụng Apache Hadoop" đã được hoàn thành với những kết quả như sau:

➢ Hiểu rõ tổng quan về big data, hadoop và MapReduce.
➢ Hiểu cấu trúc dữ liệu YouTube và ý nghĩa các thuộc tính trong tập dữ liệu.
➢ Triển khai giải pháp sử dụng Hadoop để phân tích dữ liệu YouTube.
➢ Xây dựng thành công chương trình demo cho bài toán phân tích dữ liệu.
➢ Đưa ra đánh giá và báo cáo kết quả.

Tuy nhiên, vẫn tồn tại một số hạn chế:
➢ Khả năng phân tích dữ liệu vẫn chỉ dừng lại ở một phạm vi nhỏ.
➢ Chương trình demo chưa hỗ trợ linh hoạt cho các tập dữ liệu khác.

4.2 Hướng phát triển.

Mở rộng khả năng xử lý dữ liệu lớn, phân tích đa dạng các tập dữ liệu thuộc nhiều lĩnh vực khác nhau.

➢ Tích hợp thêm các mô hình học máy, như clustering và regression, để khai thác sâu hơn giá trị của dữ liệu.

➢ Tối ưu hiệu suất và tăng tính tổng quát cho chương trình.

Trong quá trình thực hiện, nhóm đã nỗ lực tìm hiểu và ứng dụng các kiến thức mới. Tuy nhiên, thời gian có hạn khiến nhóm khó tránh khỏi thiếu sót. Rất mong nhận được ý kiến đóng góp của thầy cô và các bạn để hoàn thiện hơn trong tương lai.

TÀI LIỆU THAM KHẢO

- https://github.com/SarahAyaz/YouTube_Data_Analysis
- https://www.scaler.com/topics/hadoop/youtube-data-analysis-using-hadoop/
- https://www.youtube.com/watch?v=7E_3anKNfBw
- https://www.youtube.com/watch?v=zsViXOHtKrw
- https://github.com/rishabmenon/YouTube-Data-Analysis-Hadoop

NHIỆM VỤ CỦA CÁC THÀNH VIÊN

Họ và tên	**Công việc**
Chu Thân Nhất	+ Tìm hiểu tổng quan về map reduce + Phân tích dữ liệu USvideos0.csv + Đưa ra hướng phát triển + Chạy chương trình demo và đánh giá. + Làm báo cáo.

Lê Nguyên Vũ	+ Tìm hiểu tổng quan về hadoop. + Ý tưởng phân tích Youtube. + Cài đặt Mapper, Reducer, Partitioner + Cài đặt chương trình demo và đánh giá. + Làm báo cáo
Nguyễn Trọng Khánh	+ Tìm hiểu tổng quan về big data. + Tìm hiểu về tổng quan dữ liệu youtube + Phân tích các dữ liệu đặc trưng + Cài đặt chương trình demo. + Làm báo cáo