

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

# **ĐỒ ÁN TỐT NGHIỆP**

**Phân tích dữ liệu người dùng báo điện tử**

**Phạm Đình Thắng**

thang.pd173370@sis.hust.edu.vn

**Ngành Khoa học máy tính**

**Chuyên ngành Công nghệ phần mềm**

**Giảng viên hướng dẫn:** TS. Trần Việt Trung

\_\_\_\_\_

Chữ kí GVHD

**Khoa:** Khoa học máy tính

**Trường:** Công nghệ thông tin và Truyền thông

**HÀ NỘI, 08/2022**

# LỜI CẢM ƠN

Để hoàn thành được đồ án này, em không thể thiếu sự giúp đỡ từ thầy cô, gia đình và bạn bè. Lời đầu tiên, em xin gửi lời cảm ơn chân thành nhất đến thầy giáo TS.Trần Việt Trung đã tận tình hướng dẫn, giúp đỡ và có những định hướng giúp em tiếp cận đề tài và hoàn thiện một cách tốt nhất. Cảm ơn công ty VCCORP và anh Cường (trưởng bộ phận nơi em đang làm việc) đã tạo điều kiện để em được sử dụng một số tài nguyên phục vụ cho nghiên cứu đồ án. Con xin cảm ơn bố mẹ đã làm chỗ dựa về vật chất lẫn tinh thần trong suốt 5 năm con học đại học. Xin cảm ơn bạn bè vì đã luôn bên cạnh, động viên nhau cùng cố gắng trên con đường học tập. Cuối cùng, em xin được gửi lời cảm ơn chân thành nhất các thầy cô của Đại học Bách Khoa Hà Nội nói chung và các thầy cô ở Trường CNTT&TT nói riêng, đã dạy cho em nhiều kiến thức bổ ích giúp em hoàn thiện hơn về đạo đức lẫn chuyên môn trong suốt quá trình học tập ở ngôi trường này.

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong thời đại công nghệ thông tin và truyền thông phát triển mạnh mẽ, báo điện tử cũng đang phát triển vượt bậc. Với dịch vụ Internet tạo nên một mạng thông tin báo chí điện tử sôi động có sức thu hút hàng triệu lượt người truy cập hàng ngày, báo điện tử đã trở thành một kênh quảng cáo có quy mô khổng lồ.

Để quảng cáo hiệu quả trên nền tảng báo điện tử cần có thông tin về người dùng. Nhật ký truy cập của người dùng là một nguồn dữ liệu quan trọng trong việc cung cấp thông tin đó. Nguồn dữ liệu này có kích thước lớn, đòi hỏi các nhà phân phối quảng cáo phải có một nền tảng lưu trữ và tính toán trên dữ liệu lớn cũng như biểu diễn dữ liệu tính toán được một cách trực quan.

Để giải quyết vấn đề này, em đề xuất xây dựng một hệ thống lưu trữ-phân tích-trực quan hóa dữ liệu sử dụng các công nghệ Hadoop, Spark, Google Data Studio. Hệ thống này có ưu điểm dễ dàng mở rộng quy mô, tạo ra báo cáo phân tích đẹp với nhiều tính năng nâng cao mà không cần bỏ quá nhiều công sức để xây dựng giao diện.

Đồ án hướng tới tạo ra một mô hình hệ thống giải quyết bài toán phân tích dữ liệu người dùng nói riêng và phân tích dữ liệu lớn nói chung trong các doanh nghiệp.

# ABSTRACT

In the era of rapid development of information and communication technology, electronic newspapers are also developing rapidly. With Internet services creating a vibrant electronic press information network that attracts millions of daily visitors, e-newspapers have become a huge advertising channel.

To advertise effectively on the e-newspaper platform, information about users is required. User's access logs are an important source of data in providing such information. This data source is large, requiring ad distributors to have a big data storage and computation platform as well as to represent the computational data visually.

To solve this problem, I propose to build a data storage-analysis-visualization system using Hadoop, Spark, Google Data Studio technologies. This system has the advantage of being easy to scale, creating beautiful analytical reports with many advanced features without spending too much effort to build the interface.

The project aims to create a system model to solve the problem of user data analysis in particular and big data analysis in general in enterprises.

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án .....	2
<b>CHƯƠNG 2. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>4</b>
2.1 Dữ liệu lớn.....	4
2.2 Apache Hadoop.....	5
2.2.1 HDFS .....	5
2.2.2 YARN.....	7
2.3 Apache Spark.....	9
2.3.1 RDD.....	9
2.3.2 Spark SQL .....	10
2.3.3 Cluster Mode.....	10
2.4 Google Data Studio.....	12
<b>CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>	<b>14</b>
3.1 Nguồn dữ liệu .....	14
3.2 Sơ đồ hệ thống .....	15
3.3 Báo cáo phân tích .....	16
3.4 Cơ sở dữ liệu phân tích.....	17
<b>CHƯƠNG 4. XÂY DỰNG HỆ THỐNG .....</b>	<b>21</b>
4.1 Cài đặt cụm Hadoop/Spark.....	21
4.2 Lưu trữ dữ liệu .....	22
4.3 Phân tích dữ liệu.....	25

4.4 Trục quan hóa dữ liệu phân tích .....	33
<b>CHƯƠNG 5. KẾT QUẢ THỰC NGHIỆM.....</b>	<b>36</b>
5.1 Tích hợp .....	36
5.2 Kết quả và đánh giá .....	39
5.2.1 Chức năng chung .....	41
5.2.2 Các biểu đồ .....	42
<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>48</b>
6.1 Kết luận .....	48
6.2 Hướng phát triển.....	48
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>50</b>

## DANH MỤC HÌNH VẼ

Hình 2.1	Kiến trúc HDFS . . . . .	6
Hình 2.2	Các khối dữ liệu và bản sao phân bố trên các DataNode . . . .	7
Hình 2.3	Kiến trúc YARN . . . . .	8
Hình 2.4	Tổng quan kiến trúc Spark ở chế độ cụm . . . . .	11
Hình 2.5	Các thành phần của Google Data Studio . . . . .	13
Hình 3.1	Một bản ghi dữ liệu thô . . . . .	14
Hình 3.2	Sơ đồ tổng quan hệ thống . . . . .	15
Hình 3.3	Các bảng trong cơ sở dữ liệu phân tích . . . . .	20
Hình 4.1	Kiến trúc Spark với cụm Hadoop YARN . . . . .	21
Hình 4.2	Kết nối cơ sở dữ liệu với Google Data Studio . . . . .	33
Hình 4.3	Cấu hình nguồn dữ liệu . . . . .	33
Hình 4.4	Danh sách bảng kết nối với Google Data Studio . . . . .	34
Hình 4.5	Danh sách Blend được sử dụng . . . . .	34
Hình 5.1	Cấu hình cụm Hadoop . . . . .	36
Hình 5.2	Cấu trúc thư mục lưu trữ ở HDFS . . . . .	37
Hình 5.3	Mẫu một số bản ghi ở định dạng parquet . . . . .	37
Hình 5.4	Giao diện Web của trình quản lý tài nguyên YARN . . . . .	38
Hình 5.5	Trạng thái của một công việc Spark . . . . .	38
Hình 5.6	Giao diện tổng quan . . . . .	39
Hình 5.7	Giao diện Demographics . . . . .	40
Hình 5.8	Bộ lọc Device . . . . .	41
Hình 5.9	Bộ lọc khoảng thời gian . . . . .	41
Hình 5.10	Biểu đồ lưu lượng truy cập theo thời gian . . . . .	42
Hình 5.11	Biểu đồ lưu lượng truy cập theo thiết bị/công nghệ sử dụng . .	43
Hình 5.12	Biểu đồ lưu lượng truy cập theo vị trí địa lý . . . . .	44
Hình 5.13	Tổng quan phân bố người dùng theo độ tuổi và giới tính . . . .	45
Hình 5.14	Phân bố người dùng theo độ tuổi khi lọc theo loại thiết bị . . .	45
Hình 5.15	Lượng người dùng theo độ tuổi và giới tính hàng ngày . . . .	46
Hình 5.16	Lưu lượng truy cập theo khung giờ trong ngày . . . . .	46

## DANH MỤC BẢNG BIỂU

Bảng 3.1	Thiết kế các biểu đồ ở báo cáo đích . . . . .	17
Bảng 3.2	Bảng overview . . . . .	18
Bảng 3.3	Bảng browser_analysis . . . . .	18
Bảng 3.4	Bảng os_analysis . . . . .	18
Bảng 3.5	Bảng location_analysis . . . . .	18
Bảng 3.6	Bảng age_analysis . . . . .	19
Bảng 3.7	Bảng gender_analysis . . . . .	19
Bảng 3.8	Bảng time_frame_analysis . . . . .	19
Bảng 4.1	Cấu trúc dữ liệu lưu trữ . . . . .	22



## DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
HDFS	Hệ thống tệp phân tán Hadoop (Hadoop Distributed File System)
YARN	Trình quản lý tài nguyên của Hadoop (Yet Another Resource Negotiator)

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Mạng Internet ra đời đã mang lại rất nhiều tiện ích cho con người, từ gửi thư điện tử (e-mail) đến chuyện trò gián tiếp (message), chuyện trò trực tuyến (chat)... và một công cụ hữu dụng nhất thay cho khả năng chứa đựng, sắp xếp thông tin và huy động trí nhớ của bộ não con người đó là công cụ tìm kiếm (hàng ngày, trên thế giới có hàng triệu người cùng lúc ứng dụng công cụ tìm kiếm). Công nghệ thông tin phát triển, các phương thức cung cấp thông tin khác dần lỗi thời, cũng từ đó báo điện tử được hình thành.

Ngày nay, với lợi thế đa phương tiện, báo điện tử đã gần như thay thế hoàn toàn báo giấy, báo in truyền thống. Cũng như các kênh truyền thông khác, doanh thu chủ yếu của các trang báo điện tử đến từ quảng cáo. Số liệu thống kê của Hiệp hội Báo chí thế giới (WAN - IFRA) cho thấy, trong vòng 5 năm trở lại đây, số tiền thu về từ quảng cáo luôn chiếm từ 70 - 80% doanh thu của các báo điện tử. Và với lưu lượng truy cập của người đọc lên đến hàng triệu lượt mỗi ngày, báo điện tử trở thành một thị trường quảng cáo có quy mô khổng lồ sánh ngang với các kênh truyền hình và mạng xã hội.

Nắm bắt được đặc điểm về tập người đọc báo điện tử là yêu cầu không thể thiếu để các nhà phân phối quảng cáo đưa ra các chiến lược, giúp các chiến dịch quảng cáo tiếp cận người dùng một cách hiệu quả. Vì vậy cần có công cụ để phân tích dữ liệu người dùng và trực quan hóa kết quả phân tích để các bên liên quan có thể dễ dàng hiểu và nắm bắt được các đặc điểm đó. Đối với các trang báo điện tử, nhật ký truy cập là một nguồn dữ liệu dồi dào về người dùng, đây là một nguồn dữ liệu lớn. Do đó, bài toán đặt ra là xây dựng một hệ thống với luồng xử lý dữ liệu lớn, từ lưu trữ-tính toán cho đến trực quan hóa dữ liệu để khai thác nguồn dữ liệu này.

## 1.2 Mục tiêu và phạm vi đề tài

Hiện nay, công cụ Google Analytics của Google đang cung cấp dịch vụ miễn phí cho phép tạo ra các bảng thống kê chi tiết về khách đã ghé thăm một trang web. Phân tích Google Analytics có thể xác định các trang có hiệu suất kém bằng các kỹ thuật như hình dung kênh, nơi khách truy cập đến từ (vd: mạng xã hội, website, quảng cáo,...), họ ở lại bao lâu, vị trí địa lý và vô vàn tính năng khác. Nó cũng cung cấp nhiều tính năng nâng cao, bao gồm phân khúc khách truy cập tùy chỉnh.

Tuy nhiên Google Analytics thường chỉ phù hợp với các trang web vừa và nhỏ, có lưu lượng truy cập không quá lớn, nơi mà bên quản lý trang web khó hoặc không

muốn xây dựng công cụ phân tích của riêng mình. Bởi lẽ Google Analytics sẽ giới hạn lượng truy cập phân tích nếu trang web đó không nằm trong mạng lưới quảng cáo của Google Ads. Với những doanh nghiệp lớn hơn, có hệ thống quảng cáo của riêng mình, không chỉ vấn đề về lưu lượng truy cập lớn mà bài toán bảo mật dữ liệu luôn được đặt lên hàng đầu, họ sẽ không muốn có một bên thứ ba nào đó có thể biết và can thiệp đến dữ liệu về người dùng của mình.

Trong phạm vi đề án này, nguồn dữ liệu mà em sử dụng đến từ các trang báo điện tử như Kênh 14, CafeF ... và một số trang báo khác thuộc sở hữu của Công ty cổ phần VCCorp (nơi em đang làm việc). Xây dựng hệ thống phân tích và trực quan hóa dữ liệu nhật ký từ các trang báo điện tử này sẽ góp phần giải quyết các bài toán liên quan đến vận hành quảng cáo của công ty.

### 1.3 Định hướng giải pháp

Từ những yêu cầu đặt ra đó, em đề xuất xây dựng một hệ thống có khả năng lưu trữ và tính toán trên dữ liệu lớn, kết hợp với công cụ trực quan hóa dữ liệu phân tích. Các thành phần chính của hệ thống gồm có: (i) kho lưu trữ, (ii) chương trình phân tích dữ liệu, (iii) báo cáo phân tích.

Các công nghệ Hadoop và Spark đã chứng minh được tính hiệu quả trong việc giải quyết các bài toán liên quan đến lưu trữ, xử lý dữ liệu lớn trong những năm qua. Công cụ Google Data Studio sẽ giúp tạo báo cáo với độ tùy chỉnh cao thay vì bỏ nhiều công sức để xây dựng giao diện riêng. Những công nghệ nêu trên sẽ được sử dụng để xây dựng các thành phần tương ứng của hệ thống.

Xây dựng hệ thống theo mô hình này sẽ không chỉ giải quyết bài toán phân tích dữ liệu người đọc báo điện tử trong khuôn khổ đề án. Mở rộng ra, hệ thống có thể giải quyết các bài toán liên quan đến phân tích dữ liệu lớn nói chung.

### 1.4 Bố cục đề án

Phần còn lại của báo cáo đề án tốt nghiệp này được tổ chức như sau.

Chương 2 giới thiệu về dữ liệu lớn và các công nghệ được sử dụng trong đề án, gồm có: nền tảng lưu trữ và xử lý dữ liệu Apache Hadoop, công cụ phân tích dữ liệu Apache Spark, nền tảng tạo báo cáo Google Data Studio.

Chương 3 sẽ đi vào phân tích và thiết kế hệ thống. Từ việc thiết kế tổng quan hệ thống cho đến thiết kế chi tiết cấu trúc các biểu đồ, thiết kế các bảng cơ sở dữ liệu phân tích.

Từ thiết kế ở Chương 3, Chương 4 sẽ trình bày về việc xây dựng hệ thống. Cài đặt cụm Hadoop/Spark là bước đầu tiên. Tiếp theo là xây dựng các chương trình thực hiện lưu trữ và phân tích dữ liệu trên nền tảng đã cài đặt. Cuối cùng, xây dựng

báo cáo phân tích và kết nối với nguồn dữ liệu để trực quan hóa dữ liệu phân tích lên các biểu đồ.

Chương 5 sẽ tiến hành ghép nối các thành phần đã được xây dựng ở Chương 4 thành một hệ thống hoàn chỉnh. Các biểu đồ ở báo cáo phân tích chính là kết quả đầu ra đạt được của hệ thống, vì vậy, chương này cũng sẽ giới thiệu chi tiết từng biểu đồ cũng với nhận xét.

Chương 6, chương cuối cùng của đề án, trình bày đóng góp chính của đề án, là một mô hình hệ thống cung cấp khả năng lưu trữ-tính toán-trực quan hóa dữ liệu lớn dựa trên các công nghệ đang được sử dụng phổ biến hiện nay, hệ thống này có khả năng mở rộng. Kết luận về đề án và hướng phát triển trong tương lai cũng sẽ được trình bày trong chương này.

## CHƯƠNG 2. CÔNG NGHỆ SỬ DỤNG

### 2.1 Dữ liệu lớn

Trước khi giới thiệu về các nền tảng, công nghệ được sử dụng trong đồ án. Chúng ta nên tìm hiểu về dữ liệu lớn.

Dữ liệu lớn [1] là một thuật ngữ chỉ các tập dữ liệu khổng lồ có cấu trúc lớn, đa dạng và phức tạp với những khó khăn trong việc lưu trữ, phân tích và trực quan hóa cho các quá trình hoặc kết quả tiếp theo. Dữ liệu lớn được mô tả bởi ba đặc trưng:

- **Dung lượng (Volume):** Số lượng dữ liệu được tạo ra và lưu trữ. Kích thước của dữ liệu xác định giá trị và tiềm năng, có thể lên đến hàng terabyte thậm chí là petabyte.
- **Vận tốc (Velocity):** Trong trường hợp này nghĩa là tốc độ các dữ liệu được tạo ra và xử lý để đáp ứng các nhu cầu và thách thức trên con đường tăng trưởng và phát triển.
- **Tính đa dạng (Variety):** Các dạng và kiểu của dữ liệu. Dữ liệu được thu thập từ nhiều nguồn khác nhau và các kiểu dữ liệu cũng có rất nhiều cấu trúc khác nhau.

Trong vài năm trở lại đây, có thêm hai đặc trưng của dữ liệu lớn mà người ta quan tâm đến là **giá trị (value)** và **tính xác thực (veracity)**. Dữ liệu luôn có giá trị nhưng sẽ vô dụng nếu nó không được khám phá, đồng thời dữ liệu cũng phải đảm bảo độ tin cậy mới có thể tạo ra giá trị thật sự.

Ứng dụng của dữ liệu lớn rất đa dạng, ví dụ: phát triển sản phẩm, trải nghiệm khách hàng, học máy...

Dữ liệu lớn có rất nhiều hứa hẹn, nhưng cũng có rất nhiều thách thức. Mặc dù các công nghệ mới đã được phát triển để lưu trữ dữ liệu, nhưng khối lượng dữ liệu đang tăng gấp đôi về kích thước khoảng hai năm một lần. Các tổ chức vẫn phải vật lộn để theo kịp dữ liệu của họ và tìm cách lưu trữ nó một cách hiệu quả. Không chỉ vấn đề về lưu trữ, việc quản lý và tổ chức dữ liệu cũng tiêu tốn phần lớn thời gian của các nhà khoa học dữ liệu.

Công nghệ dữ liệu lớn đang thay đổi với tốc độ nhanh chóng. Một vài năm trước, Apache Hadoop là công nghệ phổ biến được sử dụng để xử lý dữ liệu lớn. Sau đó, Apache Spark được giới thiệu vào năm 2014. Ngày nay, sự kết hợp của hai khuôn khổ dường như là cách tiếp cận tốt nhất. Bắt kịp với công nghệ dữ liệu lớn là một thách thức không ngừng.

## 2.2 Apache Hadoop

Thư viện phần mềm Apache Hadoop [2] là một khuôn khổ cho phép xử lý phân tán các tập dữ liệu lớn trên các cụm máy tính bằng cách sử dụng các mô hình lập trình đơn giản.

Nó được thiết kế để mở rộng quy mô từ các máy chủ đơn lẻ lên hàng nghìn máy, mỗi máy đều cung cấp khả năng tính toán và lưu trữ cục bộ. Thay vì dựa vào phần cứng để cung cấp tính khả dụng cao, bản thân thư viện được thiết kế để phát hiện và xử lý các lỗi ở lớp ứng dụng, do đó, việc cung cấp dịch vụ có tính khả dụng cao trên đầu một cụm máy tính, mà mỗi máy tính đều có thể dễ bị lỗi.

Dự án Apache Hadoop bao gồm các mô-đun sau:

- **Hadoop Common:** Các tiện ích chung hỗ trợ các mô-đun Hadoop khác.
- **Hadoop Distributed File System (HDFS):** Hệ thống tệp phân tán cung cấp quyền truy cập thông lượng cao vào dữ liệu ứng dụng.
- **Hadoop YARN:** Một khuôn khổ để lập lịch công việc và quản lý tài nguyên cụm.
- **Hadoop MapReduce:** Một hệ thống dựa trên YARN để xử lý song song các tập dữ liệu lớn.

Đồ án này chủ yếu sử dụng hai mô-đun, HDFS để tổ chức lưu trữ dữ liệu, YARN sẽ điều phối các công việc và tài nguyên khi thực hiện tính toán trên dữ liệu. Chúng ta sẽ tìm hiểu sâu hơn về hai mô-đun này.

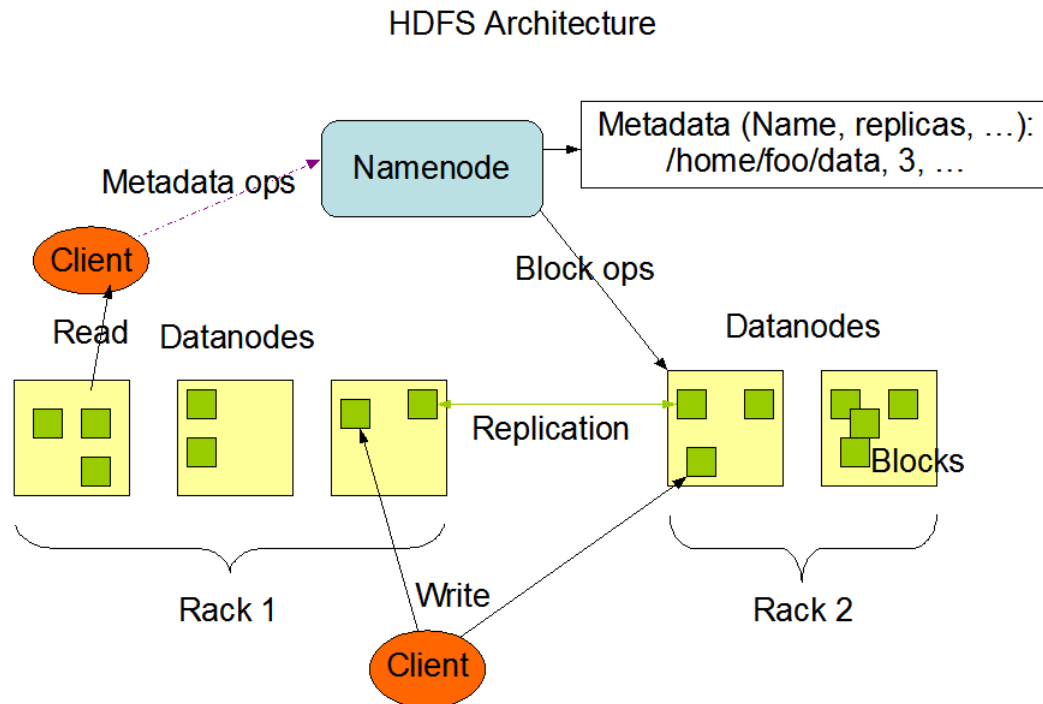
### 2.2.1 HDFS

HDFS là một hệ thống tệp phân tán được thiết kế để chạy trên phần cứng hàng hóa. Nó có nhiều điểm tương đồng với các hệ thống tệp phân tán hiện có. Tuy nhiên, sự khác biệt so với các hệ thống tệp phân tán khác là đáng kể.

HDFS có khả năng chịu lỗi cao và được thiết kế để triển khai trên phần cứng giá rẻ. HDFS cung cấp khả năng truy cập thông lượng cao vào dữ liệu ứng dụng và phù hợp với các ứng dụng có tập dữ liệu lớn. HDFS nổi lỏng một số yêu cầu POSIX để cho phép truy cập trực tuyến vào dữ liệu hệ thống tệp.

HDFS có kiến trúc master/slave. Một cụm HDFS bao gồm một NameNode duy nhất, là một máy chủ quản lý không gian tên hệ thống tệp và điều chỉnh quyền truy cập vào tệp của máy khách. Ngoài ra, có một số DataNode, thường là một DataNode cho mỗi Node trong cụm, quản lý bộ nhớ gắn liền với các Node mà chúng chạy trên đó.

HDFS để lộ không gian tên hệ thống tệp và cho phép dữ liệu người dùng được lưu trữ trong tệp. Bên trong, một tệp được chia thành một hoặc nhiều khối và các khối này được lưu trữ trong một tập hợp các DataNode. NameNode thực thi các hoạt động không gian tên của hệ thống tệp như mở, đóng và đổi tên tệp và thư mục. Nó cũng xác định ánh xạ của các khối tới DataNode. Các DataNode chịu trách nhiệm phục vụ các yêu cầu đọc và ghi từ các máy khách của hệ thống tệp. Các DataNodes cũng thực hiện việc tạo, xóa và sao chép khối theo hướng dẫn từ NameNode. Hình 2.2 minh họa kiến trúc HDFS.



**Hình 2.1:** Kiến trúc HDFS

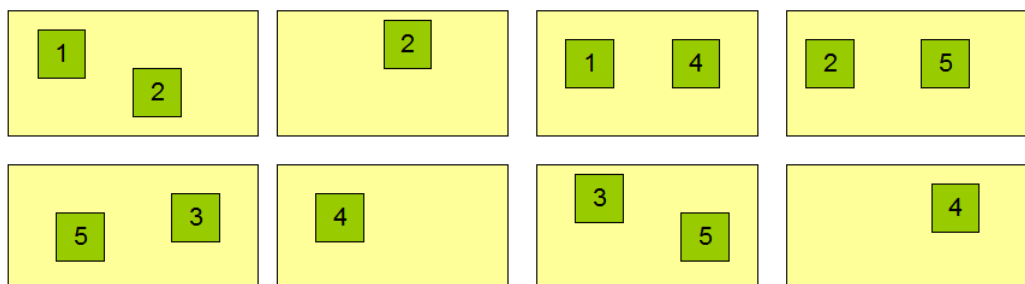
HDFS được thiết kế để lưu trữ một cách đáng tin cậy các tệp rất lớn trên các máy trong một cụm lớn. Nó lưu trữ mỗi tệp dưới dạng một chuỗi các khối. Các khối của tệp được sao chép để chịu lỗi. Kích thước khối và hệ số nhân bản có thể định cấu hình cho từng tệp. Các tệp trong HDFS được ghi một lần và chỉ có một người viết bất kỳ lúc nào.

NameNode đưa ra tất cả các quyết định liên quan đến việc nhân rộng các khối. Nó định kỳ nhận được Heartbeat and a Blockreport từ mỗi DataNode trong cụm. Việc nhận được Heartbeat ngụ ý rằng DataNode đang hoạt động bình thường. Một Blockreport chứa danh sách tất cả các khối trên DataNode.

## Block Replication

```
Namenode (Filename, numReplicas, block-ids, ...)
/users/sameerp/data/part-0, r:2, {1,3}, ...
/users/sameerp/data/part-1, r:3, {2,4,5}, ...
```

## Datanodes



**Hình 2.2:** Các khối dữ liệu và bản sao phân bố trên các DataNode

Với kiến trúc và những ưu điểm kể trên, HDFS đang được sử dụng rất rộng rãi trong việc lưu trữ dữ liệu lớn. Đó là lý do em chọn HDFS để tổ chức lưu trữ dữ liệu trong đồ án này.

### 2.2.2 YARN

Ý tưởng cơ bản của YARN là phân chia các chức năng của quản lý tài nguyên và lập lịch/giám sát công việc thành các daemon riêng biệt. Có một ResourceManager toàn cầu (RM) và ApplicationMaster cho mỗi ứng dụng (AM). Một ứng dụng là một công việc đơn lẻ hoặc một chuỗi các công việc.

ResourceManager và NodeManager tạo thành khung tính toán dữ liệu. Resource-Manager là cơ quan tối cao phân xử tài nguyên giữa tất cả các ứng dụng trong hệ thống. NodeManager là tác nhân trên mỗi máy chịu trách nhiệm về các vùng chứa, giám sát việc sử dụng tài nguyên của chúng (cpu, bộ nhớ, đĩa, mạng) và báo cáo tương tự cho ResourceManager/Scheduler.

Trên thực tế, ApplicationMaster cho mỗi ứng dụng được giao nhiệm vụ đàm phán các tài nguyên từ ResourceManager và làm việc với (các) NodeManager để thực thi và giám sát các tác vụ.

ResourceManager có hai thành phần chính: Scheduler và ApplicationsManager.

- **Scheduler** chịu trách nhiệm phân bổ tài nguyên cho các ứng dụng đang chạy khác nhau tùy thuộc vào các ràng buộc quen thuộc về dung lượng, hàng đợi, v.v. Scheduler là bộ lập lịch thuần túy theo nghĩa là nó không thực hiện giám sát hoặc theo dõi trạng thái của ứng dụng. Ngoài ra, nó không đảm bảo về

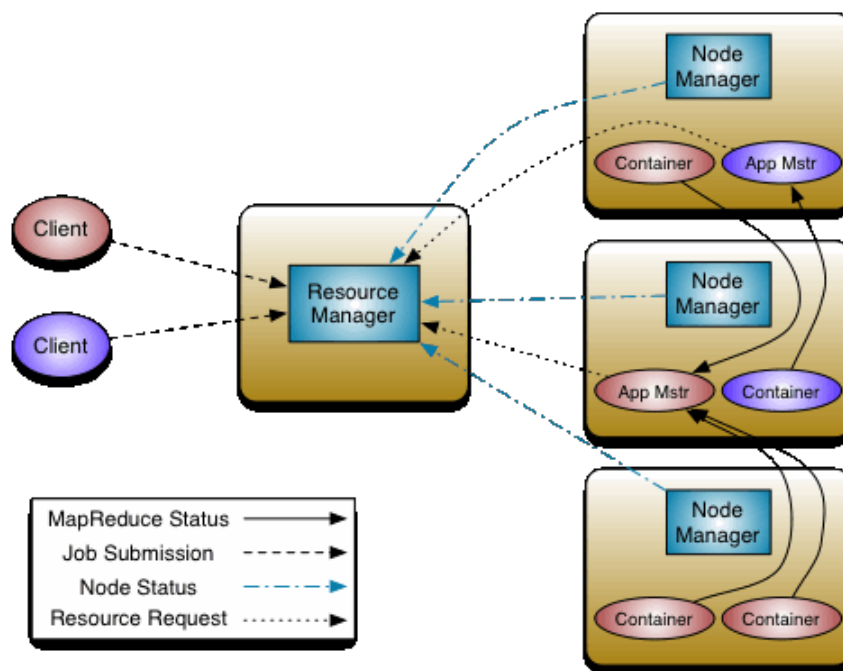


việc khởi động lại các tác vụ không thành công do lỗi ứng dụng hoặc lỗi phần cứng. Scheduler thực hiện chức năng lập lịch của nó dựa trên các yêu cầu về tài nguyên của các ứng dụng; nó làm như vậy dựa trên khái niệm trừu tượng về một vùng chứa tài nguyên kết hợp các phần tử như bộ nhớ, cpu, đĩa, mạng, v.v.

Scheduler có thể cài cắm được, hiện tại có các bộ lập lịch ví dụ như CapacityScheduler và FairScheduler.

- **ApplicationsManager** chịu trách nhiệm tiếp nhận các công việc gửi đến, thương lượng vùng chứa đầu tiên để thực thi ApplicationMaster và cung cấp dịch vụ khởi động lại vùng chứa ApplicationMaster khi bị lỗi. ApplicationMaster cho mỗi ứng dụng có trách nhiệm thương lượng các vùng chứa tài nguyên thích hợp từ Scheduler, theo dõi trạng thái của chúng và giám sát tiến trình.

Hình 2.3 minh họa kiến trúc và luồng làm việc của YARN.



**Hình 2.3:** Kiến trúc YARN

Trong đồ án này, YARN sẽ đóng vai trò quản lý tài nguyên, điều phối và lập lịch cho các công việc Spark thực hiện phân tích dữ liệu.

## 2.3 Apache Spark

Apache Spark [3] là một công cụ phân tích thống nhất để xử lý dữ liệu quy mô lớn. Nó cung cấp các API cấp cao trong Java, Scala, Python và R, và một công cụ được tối ưu hóa hỗ trợ các đồ thị thực thi chung. Nó cũng hỗ trợ một bộ công cụ cấp cao hơn phong phú bao gồm Spark SQL cho SQL và xử lý dữ liệu có cấu trúc, pandas API trên Spark cho công việc pandas, MLlib cho máy học, GraphX để xử lý đồ thị và Structured Streaming để tính toán gia tăng và xử lý luồng.

Ở cấp độ cao, mọi ứng dụng Spark đều có một chương trình điều khiển (*driver program*) chạy chương trình chính của người dùng và thực hiện các hoạt động song song khác nhau trên một cụm.

Chúng ta sẽ đi sâu hơn vào khái niệm cốt lõi của Spark - *resilient distributed dataset (RDD)*, công cụ xử lý dữ liệu có cấu trúc - Spark SQL và cơ chế hoạt động của Spark ở chế độ cụm. Đó là những kiến thức cần thiết để xây dựng đồ án.

### 2.3.1 RDD

Spark xoay quanh khái niệm về tập dữ liệu phân tán có khả năng phục hồi (RDD), là một tập hợp các phần tử có khả năng chịu lỗi có thể hoạt động song song.

RDD được tạo bằng cách bắt đầu bằng một tệp trong hệ thống tệp Hadoop (hoặc bất kỳ hệ thống tệp nào khác được Hadoop hỗ trợ) hoặc một bộ sưu tập Scala hiện có trong chương trình điều khiển và chuyển đổi nó. Người dùng cũng có thể yêu cầu Spark duy trì một RDD trong bộ nhớ, cho phép nó được sử dụng lại một cách hiệu quả trong các hoạt động song song. Cuối cùng, các RDD tự động phục hồi sau các lỗi của nút.

Các RDD hỗ trợ hai kiểu hoạt động: *transformations* (phép biến đổi), tạo tập dữ liệu mới từ tập dữ liệu hiện có và *actions* (hành động) trả về giá trị cho chương trình điều khiển sau khi chạy tính toán trên tập dữ liệu.

Tất cả các *phép biến đổi* trong Spark đều lười biếng (*lazy*), ở chỗ chúng không tính toán kết quả của chúng ngay lập tức. Thay vào đó, chúng chỉ nhớ các phép biến đổi được áp dụng cho một số tập dữ liệu cơ sở (ví dụ: một tệp). Các phép biến đổi chỉ được tính toán khi một *hành động* yêu cầu trả về kết quả cho chương trình điều khiển. Thiết kế này giúp Spark chạy hiệu quả hơn, tiết kiệm được tài nguyên.

Theo mặc định, mỗi RDD đã biến đổi cần được tính toán lại mỗi khi bạn chạy một hành động trên nó. Tuy nhiên, bạn cũng có thể duy trì một RDD trong bộ nhớ bằng cách sử dụng phương thức *persist* (hoặc *cache*), trong trường hợp này Spark sẽ giữ các phần tử gần trên cụm để truy cập nhanh hơn nhiều vào lần tiếp theo bạn

truy vấn nó. Ngoài ra còn có hỗ trợ cho các RDD duy trì trên ổ đĩa hoặc sao chép giữa các nút.

### 2.3.2 Spark SQL

Spark SQL là một mô-đun Spark để xử lý dữ liệu có cấu trúc. Không giống như API Spark RDD cơ bản, các *interface* được cung cấp bởi Spark SQL cung cấp cho Spark nhiều thông tin hơn về cấu trúc của cả dữ liệu và tính toán đang được thực hiện. Bên trong, Spark SQL sử dụng thông tin bổ sung này để thực hiện các tối ưu hóa bổ sung.

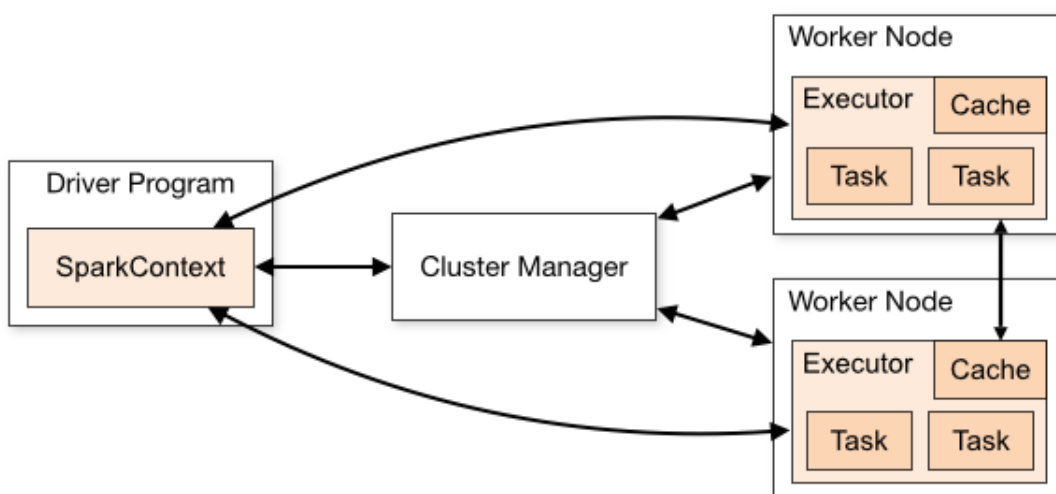
Có một số cách để tương tác với Spark SQL bao gồm *SQL* và *Dataset API*. Khi tính toán một kết quả, cùng một công cụ thực thi sẽ được sử dụng mà không phụ thuộc vào API/ngôn ngữ bạn đang sử dụng để diễn đạt tính toán. Sự thống nhất này có nghĩa là các nhà phát triển có thể dễ dàng chuyển đổi qua lại giữa các API khác nhau, dựa trên đó cung cấp cách tự nhiên nhất để thể hiện một phép biến đổi nhất định.

- **SQL:** Một công dụng của Spark SQL là thực thi các truy vấn SQL. Khi chạy SQL từ bên trong một ngôn ngữ lập trình khác, kết quả sẽ được trả về dưới dạng Dataset/DataFrame.
- **Dataset và DataFrame:**
  - Một *Dataset* là một tập hợp dữ liệu phân tán. Dataset là một interface mới được thêm vào trong Spark 1.6 cung cấp các lợi ích của RDD (kiểu chặt chẽ, khả năng sử dụng các hàm lambda mạnh mẽ) với các lợi ích của công cụ thực thi được tối ưu hóa của Spark SQL. Dataset có thể được xây dựng từ các Object JVM và sau đó được thao tác bằng cách sử dụng các phép biến đổi.
  - *DataFrame* là một Dataset được tổ chức thành các cột có tên. Về mặt khái niệm, nó tương đương với một bảng trong cơ sở dữ liệu quan hệ hoặc một khung dữ liệu trong R/Python, nhưng với các tối ưu hóa phong phú hơn. DataFrame có thể được xây dựng từ nhiều nguồn như: tệp dữ liệu có cấu trúc, bảng trong Hive, cơ sở dữ liệu bên ngoài hoặc RDD hiện có.

### 2.3.3 Cluster Mode

Các ứng dụng Spark chạy như một tập hợp các tiến trình độc lập trên một cụm, được điều phối bởi đối tượng SparkContext trong chương trình chính (được gọi là *driver program*).

Cụ thể, để chạy trên một cụm, SparkContext có thể kết nối với một số loại *trình quản lý cụm (cluster manager)*, gồm có trình quản lý cụm độc lập của Spark, Mesos, YARN hoặc Kubernetes), chúng sẽ phân bổ tài nguyên trên các ứng dụng. Một khi đã kết nối, Spark có được các *executor* trên các nút trong cụm, là các tiến trình chạy tính toán và lưu trữ dữ liệu cho ứng dụng. Tiếp theo, nó sẽ gửi mã ứng dụng (được xác định bởi các tệp JAR hoặc Python được chuyển đến SparkContext) cho các executor. Cuối cùng, SparkContext gửi các tasks đến những executor để chạy. Hình 2.4 mô tả kiến trúc của Spark ở chế độ cụm.



**Hình 2.4:** Tổng quan kiến trúc Spark ở chế độ cụm

Có một số điều hữu ích cần lưu ý về kiến trúc này:

- Mỗi ứng dụng có các tiến trình thực thi của riêng nó, các tiến trình này duy trì trong suốt thời gian của toàn bộ ứng dụng và chạy các tác vụ trong nhiều luồng. Điều này có lợi ích là cô lập các ứng dụng với nhau, ở cả phía lập lịch (mỗi trình điều khiển lập lịch cho các tác vụ riêng của mình) và phía thực thi (các tác vụ từ các ứng dụng khác nhau chạy trong các JVM khác nhau). Tuy nhiên, điều đó cũng có nghĩa là dữ liệu không thể được chia sẻ trên các ứng dụng Spark khác nhau (các phiên bản của SparkContext) mà không ghi dữ liệu đó vào hệ thống lưu trữ bên ngoài.
- Spark là bất khả tri đối với cluster manager. Miễn là nó có thể có được các tiến trình executor và các tiến trình này giao tiếp với nhau, thì việc chạy nó tương đối dễ dàng ngay cả trên một cluster manager hỗ trợ cả các ứng dụng khác (ví dụ: Mesos/YARN/Kubernetes).
- Driver program phải lắng nghe và tiếp nhận các kết nối đến từ các executor của nó trong suốt thời gian tồn tại. Như vậy, driver program phải có địa chỉ

mạng từ các worker node.

- Vì trình điều khiển lập lịch các task trên cụm, nó phải được chạy gần các worker node, tốt nhất là trên cùng một mạng cục bộ. Nếu bạn muốn gửi yêu cầu đến cụm từ xa, tốt hơn nên mở RPC tới trình điều khiển và để nó gửi các hoạt động từ gần đó hơn là chạy trình điều khiển ở xa các nút worker node.

Hệ thống hiện hỗ trợ một số trình quản lý cụm như sau:

- **Standalone:** một trình quản lý cụm đơn giản đi kèm với Spark giúp dễ dàng thiết lập một cụm.
- **Apache Mesos:** một trình quản lý cụm chung cũng có thể chạy Hadoop MapReduce và các ứng dụng dịch vụ. (Không khuyến khích dùng nữa)
- **Hadoop YARN:** trình quản lý tài nguyên trong Hadoop 2 và 3.
- **Kubernetes:** một hệ thống mã nguồn mở để tự động hóa việc triển khai, mở rộng quy mô và quản lý các ứng dụng được đóng gói.

Đề án này sử dụng Hadoop cho nên Hadoop YARN sẽ được chọn làm trình quản lý cụm cho Spark.

## 2.4 Google Data Studio

Google Data Studio là công cụ tạo dashboard báo cáo miễn phí, dễ dàng kết nối, thu thập dữ liệu từ các nguồn dữ liệu online và offline khác nhau. Với giao diện kéo thả dễ sử dụng, người dùng có thể tạo báo cáo chỉ trong vòng vài phút mà không cần phải có nhiều kiến thức gì về mặt kỹ thuật [4].

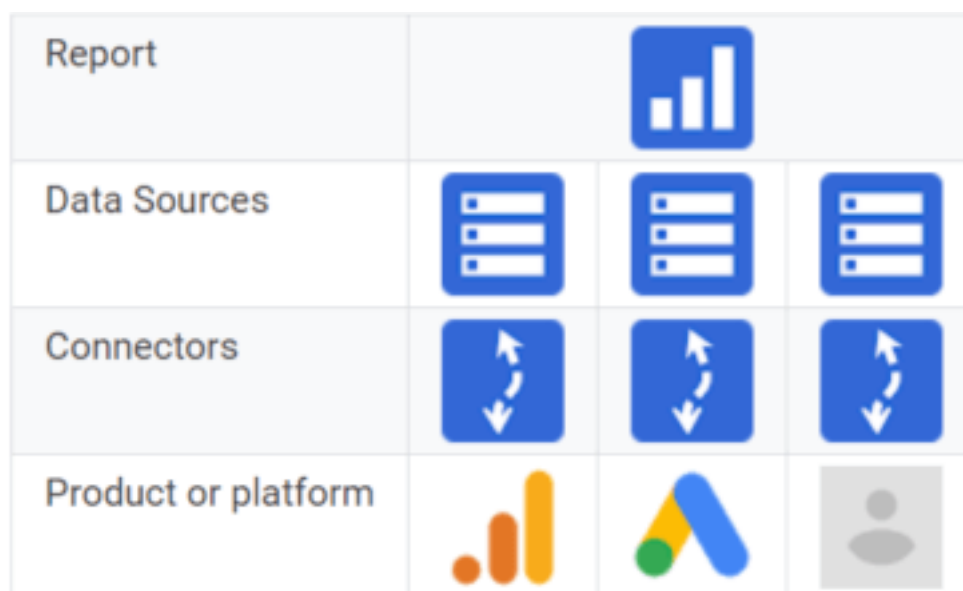
Google Data Studio có những ưu điểm sau:

- **Tạo báo cáo đa kênh:** Lấy dữ liệu từ nhiều nguồn khác nhau như Google Analytics, Google Sheets, Facebook Ads, v.v.
- **Tạo báo cáo theo thời gian thực:** Một khi đã kết nối với các nguồn dữ liệu, báo cáo của sẽ cập nhật mỗi khi có bất kỳ thay đổi nào trong nguồn dữ liệu, hoặc cập nhật tự động theo thời gian thực.
- **Thay đổi timeline báo cáo dễ dàng:** Có thể nhanh chóng chuyển đổi báo cáo tháng sang quý hoặc năm một cách nhanh chóng.
- **Giàu tính tương tác:** Người dùng có thể tương tác với báo cáo và cập nhật các trường một cách đa dạng giúp trực quan hóa dữ liệu hơn.
- **Dễ sử dụng:** Google Data Studio là một công cụ khá dễ sử dụng, dù rằng thực tế để khai thác hết công dụng của nó cũng cần có kinh nghiệm.

Google Data Studio hoạt động bằng cách kết nối báo cáo với nhiều nguồn dữ liệu khác nhau, cho phép trình bày dữ liệu trong các dashboard giàu tính tương tác.

Việc kết nối tới dữ liệu đòi hỏi phải có sự hợp tác giữa hai thành phần khác nhau.

- **Connector:** là cầu nối liên kết Google Data Studio với nền tảng của người dùng. Một khi kết nối thành công, Google sẽ tạo một *nguồn dữ liệu (data source)* trong Data Studio.
- **Data source:** là kết quả của kết nối đề cập ở trên. Ví dụ: Kết nối tới tài khoản Facebook Ads sẽ tạo ra một nguồn dữ liệu. Các nguồn dữ liệu cho phép người dùng tiếp cận dimension (thuộc tính của dữ liệu, giúp mô tả dữ liệu) và metric (các dữ liệu dạng số, có thể định lượng và dễ dàng tính toán bằng các phép tính thông thường) từ tài khoản đó.



**Hình 2.5:** Các thành phần của Google Data Studio

Nói cách khác, connector cho phép Google truy cập dữ liệu từ các sản phẩm và nền tảng khác nhau, diễn dịch chúng thành nguồn dữ liệu mà Data Studio có thể dùng trong dashboard.

Các bước sử dụng để tạo báo cáo với Google Data Studio sẽ được trình bày cùng với việc xây dựng hệ thống ở Chương 4.

## CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Chương 1 và Chương 2 đã giới thiệu tổng quan về đề tài cũng như các công nghệ được sử dụng trong đề án. Trong Chương 3 này, chúng ta sẽ đi vào phân tích và thiết kế một hệ thống dựa trên các nền tảng công nghệ đã được giới thiệu ở Chương 2 để có thể giải quyết yêu cầu bài toán đặt ra.

### 3.1 Nguồn dữ liệu

Như đã trình bày ở Chương 1, nguồn dữ liệu được sử dụng trong đề án là nhật ký ghi lại khi người dùng truy cập trang web của một số trang báo điện tử ở Việt Nam. Đây là nguồn dữ liệu thuộc về công ty VCCorp - nơi em đang làm việc và em đã được phép sử dụng dữ liệu này cho mục đích làm đề án tốt nghiệp.

Hình 3.1 là mẫu một bản ghi của dữ liệu thô. Mỗi lần người dùng truy cập vào trang web báo điện tử, một dòng nhật ký như hình dưới sẽ được tạo ra và được tập hợp lưu lại trong các file dạng văn bản.

Một dòng nhật ký tương ứng với một lượt xem của người dùng, trong đó có chứa các thông tin như thời gian truy cập, URL truy cập, các thông tin liên quan đến thiết bị sử dụng để đọc báo và thông tin cá nhân của người dùng ...

```
2022-04-15 14:29:39 2021-10-27 11:48:45 14 100.0.4896.88
10 Windows 7 249602548 5 thanhnien.vn -1 -1 / https://thanhnien.
hua-danh-du-hlv-kiatisak-gui-fan-hagl-tai-afc-champions-
league-post1448933.html 6835310125249602548 -1.-1. 0 1366x768
24 0 192.168.6.176 v;1650007779294;0;0;1;0;0;1093x500;0;1;46ad42
1650007779025;0;0;24;54;13;-1650007779025;-1650007779025
https://thanhnien.vn/ e905d86d20f1b5ea655b057f8528c85b -
1 01G0P21AF52PD08CYNE96XXS92 -1
```

**Hình 3.1:** Một bản ghi dữ liệu thô

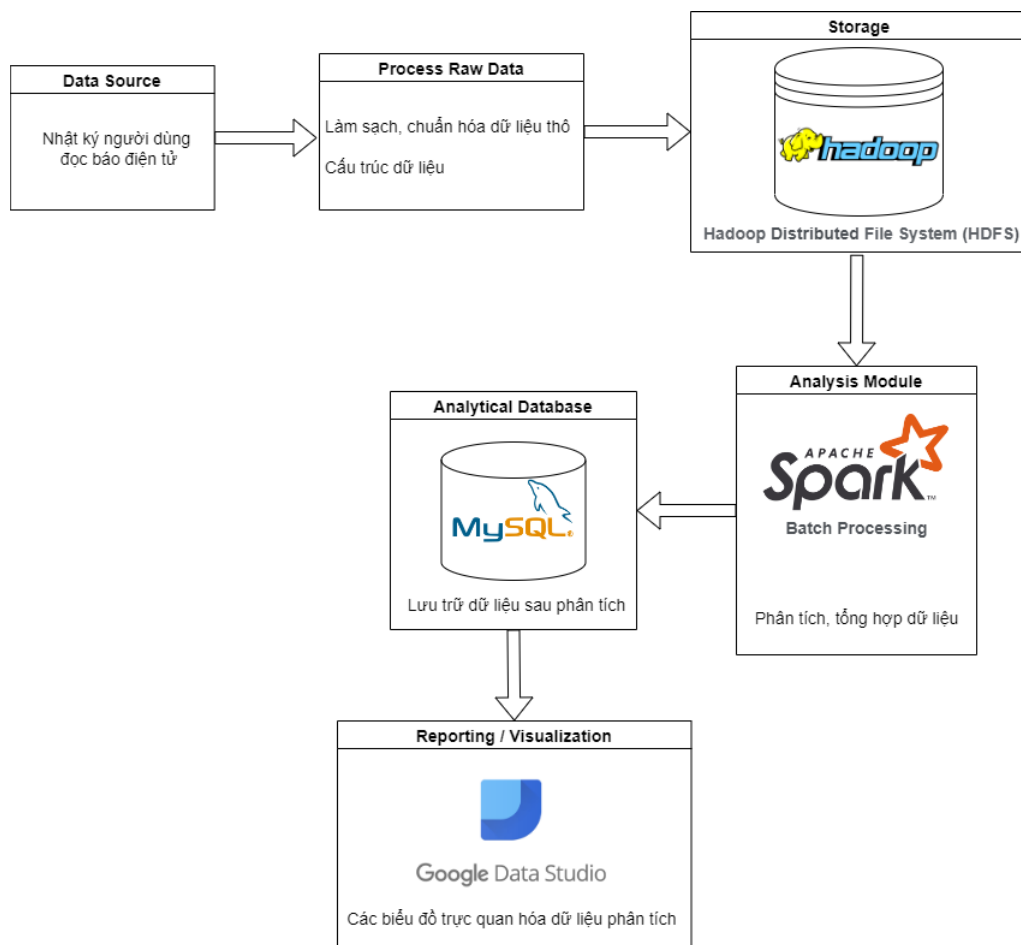
Khối lượng dữ liệu nhật ký truy cập báo điện tử của người dùng được tạo ra hàng ngày lên đến hàng Gigabyte. Đây là một nguồn dữ liệu đáng tin cậy, có dung lượng lớn, tốc độ sản sinh cao, phù hợp với khái niệm *dữ liệu lớn* (đã trình bày ở Chương 2).

### 3.2 Sơ đồ hệ thống

Dựa trên đặc điểm của nguồn dữ liệu và yêu cầu của bài toán, chúng ta cần thiết kế một hệ thống có khả năng lưu trữ và tính toán trên dữ liệu lớn. Đầu vào của hệ thống là nhật ký người dùng báo điện tử và đầu ra là báo cáo phân tích (Mục 3.3).

Sơ đồ hình 3.2 thể hiện đầy đủ các thành phần của hệ thống. Bao gồm : nguồn dữ liệu đầu vào, kho lưu trữ, các module xử lý/phân tích dữ liệu, cơ sở dữ liệu, báo cáo phân tích. Các mũi tên thể hiện luồng đi hoàn chỉnh của dữ liệu từ đầu (dữ liệu thô ban đầu) cho đến cuối (dữ liệu trực quan ở báo cáo phân tích).

Đầu tiên, dữ liệu nhật ký khi người dùng ở dạng thô sẽ được làm sạch, chuẩn hóa, chuyển thành dữ liệu có cấu trúc ở định dạng parquet và lưu ở HDFS. Tiếp theo, chương trình Spark sẽ đọc dữ liệu từ HDFS và tiến hành phân tích, kết quả phân tích sẽ được lưu vào cơ sở dữ liệu phân tích (MySQL). Cuối cùng, sử dụng công cụ Google Data Studio kết nối với cơ sở dữ liệu phân tích để trực quan hóa dữ liệu, tạo báo cáo phân tích.



**Hình 3.2:** Sơ đồ tổng quan hệ thống



### 3.3 Báo cáo phân tích

Trước khi tiến hành bước phân tích dữ liệu, chúng ta phải xác định được đầu ra mong muốn, đó chính là báo cáo phân tích. Từ đó thiết kế cơ sở dữ liệu phù hợp để lưu trữ dữ liệu phân tích.

Báo cáo phân tích là tập hợp các bảng, biểu đồ biểu diễn dữ liệu sau phân tích một cách trực quan. Các bên liên quan có thể dễ dàng nắm bắt và hiểu về các khía cạnh của tập dữ liệu người dùng báo điện tử từ báo cáo, giúp khai thác bộ dữ liệu một cách hợp lý từ đó đưa ra chiến lược kinh doanh đúng đắn.

Bảng 3.1 mô tả thiết kế sơ bộ các biểu đồ sẽ có trong báo cáo đích.

STT	Tên biểu đồ	Kiểu	Mô tả
1	Biểu đồ lưu lượng truy cập theo thời gian	Biểu đồ đường	Trục hoành thể hiện thời gian, trục tung có đơn vị là số lượt xem. Mỗi điểm dữ liệu thể hiện số lượt xem trong một khoảng thời gian. Biểu đồ này thể hiện giá trị cụ thể trong từng khoảng thời gian, sự tăng giảm theo từng giai đoạn thời gian của số lượt xem trên các trang báo điện tử.
2	Biểu đồ lưu lượng truy cập theo trình duyệt	Bảng	Bảng gồm có ba cột: tên trình duyệt, tỉ lệ người dùng, số lượt xem. Biểu đồ cho biết các trình duyệt mà người dùng sử dụng để đọc báo điện tử và phân bố lượng người dùng/lượt xem trên các trình duyệt đó.
3	Biểu đồ lưu lượng truy cập theo hệ điều hành	Biểu đồ cột	Có hai loại cột tương ứng với hai loại đơn vị là số người dùng và số lượt xem. Mỗi cột thể hiện số người dùng hoặc lượt xem trên một hệ điều hành. Biểu đồ cho biết các hệ điều hành mà người dùng sử dụng để đọc báo điện tử và phân bố lượng người dùng/lượt xem trên các hệ điều hành đó.
4	Biểu đồ lưu lượng truy cập theo loại thiết bị	Biểu đồ hình tròn	Các thiết bị sử dụng để đọc báo điện tử được phân thành hai nhóm: PC và Mobile. Biểu đồ này thể hiện sự phân bố của số người dùng hoặc lượt xem trên hai nhóm đó.

**Bảng 3.1 tiếp tục từ trang trước**

STT	Tên biểu đồ	Kiểu	Mô tả
5	Biểu đồ lưu lượng truy cập theo vị trí địa lý	Bảng	Bảng gồm có ba cột: vị trí địa lý, tỉ lệ người dùng, số lượt xem. Cột vị trí địa lý có thể là thành phố, vùng miền hoặc đất nước tùy ý muốn của người xem biểu đồ. Biểu đồ cho biết lượng người dùng đến từ đâu, phân bố lưu lượng truy cập báo điện tử theo vị trí địa lý.
6	Biểu đồ phân bố người dùng theo độ tuổi	Biểu đồ cột	Biểu đồ gồm có năm cột, mỗi cột tương ứng với một nhóm tuổi (STT 4 Bảng 4.1), thể hiện sự phân bố số người dùng theo độ tuổi.
7	Biểu đồ phân bố người dùng theo giới tính	Biểu đồ hình tròn	Biểu đồ hình tròn chia thành hai phần biểu diễn sự phân bố số người dùng theo giới tính Nam/Nữ.
8	Biểu đồ phân bố lưu lượng truy cập theo khung giờ	Biểu đồ cột	Có hai loại cột tương ứng với hai loại thiết bị PC và MB, cho biết người dùng thường đọc báo điện tử vào khung giờ nào trong ngày, xu hướng đó phụ thuộc vào thiết bị sử dụng như thế nào .

**Bảng 3.1:** Thiết kế các biểu đồ ở báo cáo đích

### 3.4 Cơ sở dữ liệu phân tích

Sau khi xác định được các biểu đồ mong muốn ở báo cáo phân tích, việc cần làm tiếp theo là thiết kế các bảng trong cơ sở dữ liệu phân tích. Cơ sở dữ liệu phân tích lưu trữ dữ liệu đầu ra của bước phân tích dữ liệu và đóng vai trò làm nguồn dữ liệu để hiển thị lên các biểu đồ ở báo cáo.

Để phù hợp với yêu cầu bài toán, dữ liệu sau phân tích sẽ được lưu dưới dạng bảng (gồm các hàng và cột) cho nên em chọn hệ quản trị cơ sở dữ liệu MySQL. Sau đây là thiết kế chi tiết các bảng có trong cơ sở dữ liệu phân tích.

Bảng *overview* lưu trữ dữ liệu phân tích lưu lượng truy cập theo thời gian và loại thiết bị, là nguồn dữ liệu cho biểu đồ 1 và biểu đồ 4 (bảng 3.1).

STT	Tên trường	Kiểu dữ liệu	Khóa	Mô tả
1	time	datetime	PRI	Thời gian
2	device	varchar(2)	PRI	Loại thiết bị
3	user	int		Số người dùng
4	view	int		Số lượt xem

**Bảng 3.2:** Bảng overview

Bảng *browser\_analysis* lưu trữ dữ liệu phân tích lưu lượng truy cập theo trình duyệt, là nguồn dữ liệu cho biểu đồ 2 (bảng 3.1).

STT	Tên trường	Kiểu dữ liệu	Khóa	Mô tả
1	time	date	PRI	Thời gian
2	browser_id	tinyint	PRI	Mã trình duyệt
3	device	varchar(2)	PRI	Loại thiết bị
4	user	int		Số người dùng
5	view	int		Số lượt xem

**Bảng 3.3:** Bảng browser\_analysis

Bảng *os\_analysis* lưu trữ dữ liệu phân tích lưu lượng truy cập theo hệ điều hành, là nguồn dữ liệu cho biểu đồ 3 (bảng 3.1).

STT	Tên trường	Kiểu dữ liệu	Khóa	Mô tả
1	time	date	PRI	Thời gian
2	os_id	tinyint	PRI	Mã hệ điều hành
3	device	varchar(2)	PRI	Loại thiết bị
4	user	int		Số người dùng
5	view	int		Số lượt xem

**Bảng 3.4:** Bảng os\_analysis

Bảng *location\_analysis* lưu trữ dữ liệu phân tích lưu lượng truy cập theo vị trí địa lý của người dùng, là nguồn dữ liệu cho biểu đồ 5 (bảng 3.1).

STT	Tên trường	Kiểu dữ liệu	Khóa	Mô tả
1	time	date	PRI	Thời gian
2	loc_id	int	PRI	Mã vị trí
3	user	int		Số người dùng
4	view	int		Số lượt xem

**Bảng 3.5:** Bảng location\_analysis

Bảng *age\_analysis* lưu trữ dữ liệu phân tích lưu lượng truy cập theo độ tuổi, là nguồn dữ liệu cho biểu đồ 6 (bảng 3.1).

STT	Tên trường	Kiểu dữ liệu	Khóa	Mô tả
1	time	date	PRI	Thời gian
2	age	tinyint	PRI	Mã độ tuổi
3	device	varchar(2)	PRI	Loại thiết bị
4	user	int		Số người dùng
5	view	int		Số lượt xem

**Bảng 3.6:** Bảng *age\_analysis*

Bảng *gender\_analysis* lưu trữ dữ liệu phân tích lưu lượng truy cập theo giới tính, là nguồn dữ liệu cho biểu đồ 7 (bảng 3.1).

STT	Tên trường	Kiểu dữ liệu	Khóa	Mô tả
1	time	date	PRI	Thời gian
2	gender	tinyint	PRI	Mã giới tính
3	device	varchar(2)	PRI	Loại thiết bị
4	user	int		Số người dùng
5	view	int		Số lượt xem

**Bảng 3.7:** Bảng *gender\_analysis*

Bảng *time\_frame\_analysis* lưu trữ dữ liệu phân tích lưu lượng truy cập theo khung giờ trong ngày, là nguồn dữ liệu cho biểu đồ 8 (bảng 3.1).

STT	Tên trường	Kiểu dữ liệu	Khóa	Mô tả
1	time	date	PRI	Thời gian
2	device	varchar(2)	PRI	Loại thiết bị
3	frame_id	tinyint	PRI	Mã khung giờ
4	user	int		Số người dùng
5	view	int		Số lượt xem

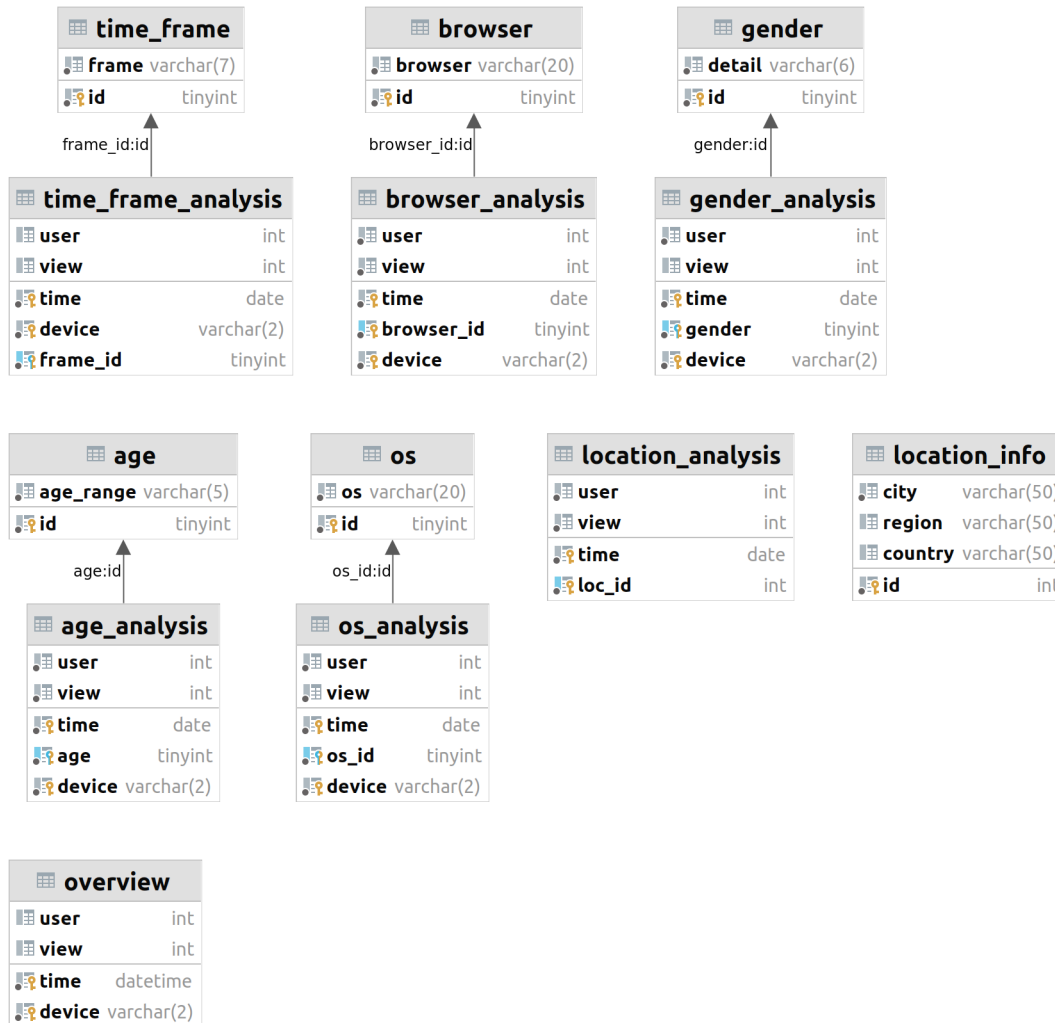
**Bảng 3.8:** Bảng *time\_frame\_analysis*

Toàn bộ các bảng chứa dữ liệu phân tích đều có trường *time* cho biết số liệu tổng hợp được sau phân tích là của thời gian nào. Dữ liệu sẽ được phân tích và tổng hợp theo đơn vị ngày, mục đích là để thuận tiện cho việc lọc và tổng hợp dữ liệu theo khoảng thời gian trên báo cáo đích, người xem báo cáo có thể thoải mái xem số liệu của một khoảng thời gian tùy ý.

Ngoài ra, loại thiết bị người dùng sử dụng để đọc báo cũng có ảnh hưởng đến số liệu phân tích được. Vì vậy, một số bảng sẽ có thêm trường *device* để nếu cần thiết

người xem báo cáo có thể quan sát và so sánh số liệu giữa Mobile và PC trên cùng một khía cạnh phân tích.

Ngoài các bảng lưu trữ dữ liệu phân tích được mô tả ở trên, còn cần các bảng lưu trữ các dữ liệu cố định để cung cấp thêm thông tin khi xuất dữ liệu từ cơ sở dữ liệu lên các biểu đồ. Ví dụ: mã vị trí 4 tương ứng với tỉnh-vùng miền-đất nước nào, mã trình duyệt 16 là trình duyệt nào... Hình 3.3 là toàn bộ các bảng thực sự có trong cơ sở dữ liệu phân tích, cấu trúc và mối quan hệ giữa các bảng.



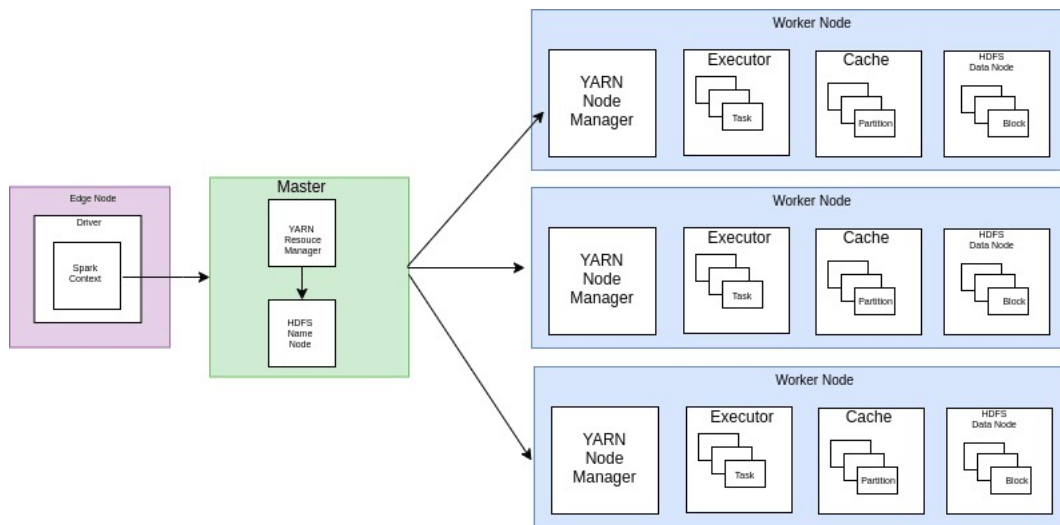
**Hình 3.3:** Các bảng trong cơ sở dữ liệu phân tích

Chương này đã thảo luận về đặc điểm của nguồn dữ liệu và đưa ra thiết kế tổng quan hệ thống: (i) sử dụng HDFS để lưu trữ dữ liệu lớn, (ii) phân tích dữ liệu lớn với Spark, (iii) trực quan hóa dữ liệu phân tích trên nền tảng Google Data Studio; Đồng thời thiết kế chi tiết báo cáo phân tích, cơ sở dữ liệu phân tích. Trên cơ sở đó, ở Chương 4 chúng ta sẽ từng bước xây dựng các thành phần của hệ thống theo thiết kế đã đề ra.

## CHƯƠNG 4. XÂY DỰNG HỆ THỐNG

Dựa vào thiết kế ở chương 3, trong chương này, chúng ta sẽ xây dựng các thành phần của hệ thống gồm có: (i) Cụm Hadoop/Spark , (ii) Chương trình lưu trữ dữ liệu , (iii) Chương trình phân tích dữ liệu và (iv) Báo cáo phân tích.

### 4.1 Cài đặt cụm Hadoop/Spark



**Hình 4.1:** Kiến trúc Spark với cụm Hadoop YARN

Cụm Hadoop được cài đặt trên ba server, gồm một Master và hai Worker, mỗi server có cấu hình 16GB RAM và bộ nhớ 256GB.

Master node có địa chỉ IP 10.5.92.76, lưu trữ metadata về hệ thống tệp phân tán và lập lịch/phân bổ tài nguyên. Master node chạy hai daemon gồm: (i) Namenode, quản lý hệ thống tệp phân tán, là đầu mối truy cập và thực hiện các thao tác với file từ phía client. Namenode có nhiệm vụ duy trì và quản lý các datanode, lưu trữ và cập nhật các metadata, ví dụ như địa chỉ của các block trên datanode, quyền truy cập của client; (ii) Resource Manager đảm nhận việc lập lịch và thực thi các tiến trình chạy trên các worker node thông qua các Node Manager.

Hai Worker node có địa chỉ IP lần lượt là 10.5.92.80 và 10.5.92.84, lưu trữ dữ liệu được đưa vào HDFS và cung cấp sức mạnh xử lý để chạy các công việc. Mỗi Worker node chạy hai daemon: (i) Datanode quản lý dữ liệu vật lý được lưu trữ trên node; (ii) Node Manager quản lý việc thực thi các tác vụ trên node.

Khi một chương trình được phía client (ví dụ Spark) gửi đến, luồng thực thi sẽ như sau:

1. Resource Manager (RM) sẽ cấp phát một vùng chứa cần thiết trên một Worker để khởi chạy Application Master (AM).
2. AM giao tiếp với RM để yêu cầu cung cấp tài nguyên để chạy chương trình.
3. Sau khi nhận thành công các vùng chứa, AM thực thi mã chương trình của người dùng thông qua (các) Node Manager (NM).
4. NM cung cấp thông tin (giai đoạn thực thi, trạng thái) của chương trình cho AM.
5. Trong thời gian chạy của chương trình người dùng, client tương tác với AM để lấy trạng thái.
6. Khi chương trình hoàn thành và tất cả các công việc cần thiết được hoàn tất, AM chấm dứt và giải phóng vùng chứa.

#### 4.2 Lưu trữ dữ liệu

Trong khuôn khổ đề án này, em sẽ không khai thác hết toàn bộ thông tin có trong dữ liệu thô, vì vậy cần có bước tiền xử lý để chọn lọc các trường dữ liệu cần thiết và chuẩn hóa định dạng của chúng.

Dữ liệu thô sẽ được chuyển thành định dạng parquet và lưu trữ trên hệ thống HDFS (đã được cài đặt ở phần trước), cấu trúc các trường mô tả ở bảng 4.1 dưới.

STT	Tên trường	Kiểu dữ liệu	Ý nghĩa
1	time	String	thời điểm truy cập trang báo
2	guid	Long	mã định danh người dùng
3	gender	Integer	mã giới tính - 1 là nam, 2 là nữ
4	age	Integer	mã độ tuổi - 3: dưới 18 tuổi - 4: từ 18 đến 24 tuổi - 5: từ 25 đến 35 tuổi - 6: từ 36 đến 50 tuổi - 7: trên 60 tuổi
5	os	Integer	mã trình duyệt sử dụng
6	browser	Integer	mã hệ điều hành thiết bị
7	location	Integer	mã vị trí địa lý của người dùng, tương ứng với tỉnh/thành phố

**Bảng 4.1:** Cấu trúc dữ liệu lưu trữ

Dưới đây là các hàm xử lý chính của chương trình thực hiện tiền xử lý và lưu trữ dữ liệu, viết bằng ngôn ngữ Java.

Hàm *extractData* nhận đầu vào là một dòng nhật ký, có nhiệm vụ trích xuất các trường dữ liệu cần thiết và trả về dưới dạng một bản ghi có cấu trúc như ở bảng 3.1

---

```
//LogProcessor.java
public GenericData.Record extractData(String line) {
    String[] s = line.split("\t");
    if (s.length >= 24) {
        GenericData.Record record = new
            GenericData.Record(SCHEMA);
        record.put("time", s[0]);
        record.put("browser", Common.IntStr(s[2]));
        record.put("os", Common.IntStr(s[4]));
        record.put("loc", Common.IntStr(s[7]));
        record.put("domain", s[8]);
        record.put("path", s[11]);
        record.put("guid", Common.LongStr(s[13].replace("[",
            "")));
        record.put("category", s[23]);
        return record;
    }
    return null;
}
```

---

Hàm *parseRawToSchema* đọc từng dòng nhật ký trong file dữ liệu thô và gọi đến hàm *extractData* để xử lý, tập hợp các bản ghi thành một danh sách.

---

```
//LogProcessor.java
public List<GenericData.Record> parseRawToSchema(String
    path, boolean isPC) {
    List<GenericData.Record> parquet = new ArrayList<>();
    try (BufferedReader br = new BufferedReader(new
        FileReader(path))) {
        String line;
        while ((line = br.readLine()) != null) {
            GenericData.Record record;
            if (isPC) {
                record = extractDataPC(line);
            } else {
                record = extractDataMB(line);
            }
        }
    }
}
```



```
        }
        if (record != null) parquet.add(record);
    }
} catch (IOException e) {
    e.printStackTrace();
}
return parquet;
}
```

---

Hàm *writeParquetFile* nhận đầu vào là một danh sách bản ghi có cấu trúc và một đường dẫn, thực hiện lưu dữ liệu ra file parquet theo đường dẫn đã cho ở HDFS.

---

```
//LogProcessor.java
public void writeParquetFile(List<GenericData.Record>
    recordsToWrite, String path) {
    Path fileToWrite = new Path(path);
    try (ParquetWriter<GenericData.Record> writer =
        AvroParquetWriter
            .<GenericData.Record>builder(fileToWrite)
            .withSchema(SCHEMA)
            .withConf(HADOOP_CONFIG)
            .withCompressionCodec(CompressionCodecName.SNAPPY)
            .build()) {

        for (GenericData.Record record : recordsToWrite) {
            writer.write(record);
        }
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

---

Hàm *buildParquet* lần lượt đọc các file dữ liệu thô theo danh sách đường dẫn truyền vào, sử dụng hàm *parseRawToSchema*. Khi gom đủ một số lượng bản ghi nhất định, hàm này gọi đến hàm *writeParquetFile* để ghi dữ liệu ra file ở HDFS.

---

```
//LogProcessor.java
public void buildParquet(String outputPath, List<String>
    filesInput, boolean isPC) throws IOException {
    List<GenericData.Record> recordsToWrite = new
        ArrayList<>();
```

```
int numFileRead = 0;
int numFileWrite = 0;
for(String file : filesInput) {
    recordsToWrite.addAll(parseRawToSchema(file, isPC));
    if(++numFileRead == 100) {
        numFileRead = 0;
        numFileWrite++;
        writeParquetFile(recordsToWrite,
            outputPath+"/"+numFileWrite+".parquet");
        recordsToWrite.clear();
    }
}
if (!recordsToWrite.isEmpty()) {
    numFileWrite++;
    writeParquetFile(recordsToWrite,
        outputPath+"/"+numFileWrite+".parquet");
}
}
```

---

### 4.3 Phân tích dữ liệu

Chúng ta đã chuẩn bị được dữ liệu đầu vào và xác định đầu ra mong muốn của bước phân tích. Tiếp theo, chúng ta cần định nghĩa các bài toán phân tích, đưa ra cách giải quyết, sau đó viết chương trình sử dụng mã Spark (Java) để giải quyết các vấn đề đó.

Các phép phân tích đều cùng đọc dữ liệu từ HDFS với cấu trúc cho trước, nhưng mỗi loại phân tích chỉ sử dụng một số trường dữ liệu cho nên trước mỗi bước phân tích đều có bước lọc chỉ giữ lại các trường dữ liệu cần thiết. Chi tiết về các bài toán phân tích sẽ được trình bày sau đây.

#### **Phân tích lưu lượng truy cập tổng quan theo thời gian**

- Bài toán: Tính số lượng người dùng, lượt xem báo điện tử hàng giờ.
- Đầu vào:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	time	String	Thời gian truy cập, ví dụ "2022-04-15 14:29:39"
2	guid	Long	Mã định danh người dùng

- Đầu ra:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	hour	String	Thời gian phân tích, đến hàng giờ, ví dụ "2022-04-15 14"
2	user	Integer	Số người dùng
3	view	Integer	Số lượt xem

- Xử lý: Cần xử lý xâu thời gian lấy đến hàng giờ, sau đó tổng hợp số liệu theo giờ. Số lượt xem sẽ là số lượng bản ghi, số người dùng là số lượng bản ghi không lặp lại dựa trên trường *guid*.
- Mã nguồn:

```
//Analyst.java
public List<Row> overview(Dataset<Row> df) {
    return df.select(substring(col("time"), 1,
        13).as("hour"),
        col("guid"))
        .groupBy("hour")
        .agg(count("guid").as("view"),
            countDistinct("guid").as("user"))
        .collectAsList();
}
```

### Phân tích lưu lượng truy cập theo trình duyệt

- Bài toán: Tính số lượng người dùng, lượt xem báo điện tử theo trình duyệt hàng ngày.
- Đầu vào:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	time	String	Thời gian truy cập, ví dụ "2022-04-15 14:29:39"
2	guid	Long	Mã định danh người dùng
3	browser	Integer	Mã trình duyệt

- Đầu ra:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	date	String	Ngày phân tích, ví dụ "2022-04-15"
2	browser	Integer	Mã trình duyệt
3	user	Integer	Số người dùng
4	view	Integer	Số lượt xem

- Xử lý: Cần xử lý chuỗi thời gian lấy đến ngày, sau đó tổng hợp số liệu theo ngày và nhóm theo trình duyệt. Số lượt xem sẽ là số lượng bản ghi, số người dùng là số lượng bản ghi không lặp lại dựa trên trường *guid*.

- Mã nguồn:

```
//Analyst.java
public List<Row> browser(Dataset<Row> df) {
    return df.select(split(col("time"), "
    ").getItem(0).as("date"),
        col("browser"), col("guid"))
        .groupBy("date", "browser")
        .agg(count("guid").as("view"),
            countDistinct("guid").as("user"))
        .collectAsList();
}
```

### Phân tích lưu lượng truy cập theo hệ điều hành

- Bài toán: Tính số lượng người dùng, lượt xem báo điện tử theo hệ điều hành hàng ngày.
- Đầu vào:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	time	String	Thời gian truy cập, ví dụ "2022-04-15 14:29:39"
2	guid	Long	Mã định danh người dùng
3	os	Integer	Mã hệ điều hành

- Đầu ra:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	date	String	Ngày phân tích, ví dụ "2022-04-15"
2	os	Integer	Mã hệ điều hành
3	user	Integer	Số người dùng
4	view	Integer	Số lượt xem

- Xử lý: Cần xử lý xâu thời gian lấy đến ngày, sau đó tổng hợp số liệu theo ngày và nhóm theo hệ điều hành. Số lượt xem sẽ là số lượng bản ghi, số người dùng là số lượng bản ghi không lặp lại dựa trên trường *guid*.

- Mã nguồn:

```
//Analyst.java
public List<Row> os(Dataset<Row> df) {
    return df.select(split(col("time"), "
    ").getItem(0).as("date"),
        col("os"), col("guid"))
        .groupBy("date", "os")
        .agg(count("guid").as("view"),
            countDistinct("guid").as("user"))
        .collectAsList();
}
```

### Phân tích lưu lượng truy cập theo khung giờ trong ngày

- Bài toán: Tính số lượng người dùng, lượt xem báo điện tử theo khung giờ trong ngày.
- Đầu vào:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	time	String	Thời gian truy cập, ví dụ "2022-04-15 14:29:39"
2	guid	Long	Mã định danh người dùng

- Đầu ra:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	date	String	Ngày phân tích, ví dụ "2022-04-15"
2	hour	Integer	Giờ phân tích, ví dụ "14"
3	user	Integer	Số người dùng
4	view	Integer	Số lượt xem

- Xử lý: Trích xuất ngày và giờ từ trường *time* để tạo ra hai trường *date* và *hour*. Tổng hợp số liệu nhóm theo *date* và *hour*, số lượt xem sẽ là số lượng bản ghi, số người dùng là số lượng bản ghi không lặp lại dựa trên trường *guid*.

- Mã nguồn:

```
//Analyst.java
public List<Row> timeframe(Dataset<Row> df) {
    return df.select(split(col("time"), "
").getItem(0).as("date"),
        substring(col("time"), 12,
            2).cast("int").as("hour"),
        col("guid"))
        .groupBy("date", "hour")
        .agg(count("guid").as("view"),
            countDistinct("guid").as("user"))
        .collectAsList();
}
```

### Phân tích lưu lượng truy cập theo vị trí địa lý

- Bài toán: Tính số lượng người dùng, lượt xem báo điện tử theo vị trí địa lý.
- Đầu vào:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	time	String	Thời gian truy cập, ví dụ "2022-04-15 14:29:39"
2	guid	Long	Mã định danh người dùng
3	loc	Integer	Mã vị trí địa lý

- Đầu ra:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	date	String	Ngày phân tích, ví dụ "2022-04-15"
2	loc	Integer	Mã vị trí địa lý
3	user	Integer	Số người dùng
4	view	Integer	Số lượt xem

- Xử lý: Trích xuất ngày từ trường *time*, loại bỏ những bản ghi có *loc*=-1 là những lượt truy cập không xác định được vị trí địa lý. Tổng hợp số liệu nhóm theo *date* và *loc*, số lượt xem sẽ là số lượng bản ghi, số người dùng là số lượng bản ghi không lặp lại dựa trên trường *guid*.
- Mã nguồn:

```
//Analyst.java
public List<Row> location(Dataset<Row> df) {
    return df.select("date", "loc", "guid")
        .filter("loc != -1")
        .groupBy("date", "loc")
        .agg(count("guid").as("view"),
            countDistinct("guid").as("user"))
        .collectAsList();
}
```

### Phân tích lưu lượng truy cập theo độ tuổi

- Bài toán: Tính số lượng người dùng, lượt xem báo điện tử theo độ tuổi.
- Đầu vào:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	time	String	Thời gian truy cập, ví dụ "2022-04-15 14:29:39"
2	guid	Long	Mã định danh người dùng
3	age	Integer	Mã nhóm tuổi

- Đầu ra:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	date	String	Ngày phân tích, ví dụ "2022-04-15"
2	age	Integer	Mã nhóm tuổi
3	user	Integer	Số người dùng
4	view	Integer	Số lượt xem

- Xử lý: Trích xuất ngày từ trường *time*, loại bỏ những bản ghi có *age* = -1 là những người dùng không xác định được độ tuổi. Tổng hợp số liệu nhóm theo *date* và *age*, số lượt xem sẽ là số lượng bản ghi, số người dùng là số lượng bản ghi không lặp lại dựa trên trường *guid*.

- Mã nguồn:

```
//Demographics.java
public List<Row> age(Dataset<Row> df) {
    return df.select(split(col("dt"), "
    ").getItem(0).as("date"),
        col("age"),
        col("guid"))
        .filter("age != -1")
        .groupBy("date", "age")
        .agg(count("guid").as("view"),
            countDistinct("guid").as("user"))
        .collectAsList();
}
```

### Phân tích lưu lượng truy cập theo giới tính

- Bài toán: Tính số lượng người dùng, lượt xem báo điện tử theo giới tính.
- Đầu vào:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	time	String	Thời gian truy cập, ví dụ "2022-04-15 14:29:39"
2	guid	Long	Mã định danh người dùng
3	gender	Integer	Mã giới tính



- Đầu ra:

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	date	String	Ngày phân tích, ví dụ "2022-04-15"
2	gender	Integer	Mã giới tính
3	user	Integer	Số người dùng
4	view	Integer	Số lượt xem

- Xử lý: Trích xuất ngày từ trường *time*, loại bỏ những bản ghi có *gender* = -1 là những người dùng không xác định được giới tính. Tổng hợp số liệu nhóm theo *date* và *gender*, số lượt xem sẽ là số lượng bản ghi, số người dùng là số lượng bản ghi không lặp lại dựa trên trường *guid*.

- Mã nguồn:

---

```
//Demographics.java
public List<Row> gender(Dataset<Row> df) {
    return df.select(split(col("dt"), "
    ").getItem(0).as("date"),
        col("gender"),
        col("guid"))
        .filter("gender != -1")
        .groupBy("date", "gender")
        .agg(count("guid").as("view"),
            countDistinct("guid").as("user"))
        .collectAsList();
}
```

---

Kết quả của các phép phân tích sẽ được lưu vào các bảng MySQL tương ứng ở cơ sở dữ liệu phân tích (thiết kế ở Chương 3).

#### 4.4 Trực quan hóa dữ liệu phân tích

Trong đồ án này, em sử dụng công cụ Google Data Studio (đã được giới thiệu trong Chương 2) để trực quan hóa dữ liệu phân tích. Các bước thực hiện như sau: (i) tạo nguồn dữ liệu, (ii) tạo báo cáo, (iii) tạo các biểu đồ trên báo cáo.

Đầu tiên chúng ta thực hiện kết nối cơ sở dữ liệu phân tích để làm nguồn dữ liệu cho Google Data Studio (hình 4.2).

The screenshot shows the Google Data Studio interface. On the left, the 'BASIC' tab is active, displaying 'Database Authentication' settings. The 'JDBC URL' section includes fields for 'Host Name or IP' (0.tcp.ap.ngrok.io), 'Port (Optional)' (11033), 'Database' (user\_view\_analysis), 'Username' (thangpd), and 'Password'. There is an 'Enable SSL' checkbox and an 'AUTHENTICATE' button. On the right, the 'TABLES' tab is active, showing a list of tables under the 'CUSTOM QUERY' section. The tables listed are: overview, age, age\_analysis, browser, browser\_analysis, gender, gender\_analysis, location\_analysis, location\_info, os, os\_analysis, time\_frame, and time\_frame\_analysis.



























**Hình 4.2:** Kết nối cơ sở dữ liệu với Google Data Studio

Sau khi kết nối nguồn dữ liệu chúng ta có thể cấu hình bảng dữ liệu để phù hợp cho việc hiển thị lên các biểu đồ, các cấu hình gồm có: tên trường, kiểu dữ liệu, phép tổng hợp mặc định.

Field ↓	Type ↓	Default Aggregation ↓	Description ↓
DIMENSIONS (4)			
Device	ABC Text	None	
Time	Date & Time (YYYYMMDDhhmmss)	None	
Users	123 Number	Sum	
Views	123 Number	Sum	
METRICS (1)			
Record Count	123 Number	Auto	

**Hình 4.3:** Cấu hình nguồn dữ liệu

Hình 4.4 là danh sách các bảng được kết nối với Google Data Studio.

Data sources			
Name	Connector Type	Type	Used in report
 MySQL - overview <a href="#">[?]</a>	MySQL	 Reusable	8 charts
 MySQL - age	SQL	 Embedded	2 charts
 MySQL - browser_info	SQL	 Embedded	1 chart
 MySQL - gender	SQL	 Embedded	2 charts
 MySQL - timeframe	SQL	 Embedded	1 chart
 MySQL - location_info	SQL	 Embedded	1 chart
 MySQL - location	SQL	 Embedded	1 chart
 MySQL - age_info	SQL	 Embedded	2 charts
 MySQL - os	SQL	 Embedded	1 chart
 MySQL - tf_info	SQL	 Embedded	1 chart
 MySQL - browser	SQL	 Embedded	1 chart
 MySQL - gender_info	SQL	 Embedded	2 charts
 MySQL - os_info	SQL	 Embedded	1 chart

**Hình 4.4:** Danh sách bảng kết nối với Google Data Studio

Ngoài các bảng lưu dữ liệu từ bước phân tích, còn cần thêm các bảng thông tin phụ để có thể diễn giải thông tin lên các biểu đồ một cách đầy đủ nhất. Google Data Studio cung cấp một khái niệm **Blend** cho phép kết nối hai hoặc nhiều nguồn dữ liệu với nhau (tương tự phép JOIN trong SQL) và có thể sử dụng nguồn dữ liệu kết hợp đó để nạp vào các biểu đồ. Chúng ta sẽ tạo ra các **Blend** từ các bảng dữ liệu phân tích và các bảng thông tin phụ (hình 4.5)

Blends	
Name	Used in report
location_full	1 chart
timeframe_full	1 chart
browser_full	1 chart
os_full	1 chart
age_full	2 charts
gender_full	2 charts

**Hình 4.5:** Danh sách Blend được sử dụng

Bước tiếp theo là tạo báo cáo. Báo cáo có thể được tạo bằng cách xây dựng thủ công các biểu đồ, hoặc chọn mẫu có sẵn thích hợp rồi tùy chỉnh.

Cuối cùng là tạo các biểu đồ, gồm các bước: (i) chọn loại biểu đồ, (ii) chọn nguồn dữ liệu (một trong các nguồn dữ liệu hoặc Blend), (iii) cấu hình biểu đồ (hình dạng, bố cục, dữ liệu). Để tránh trùng lặp về nội dung, phần này em sẽ không đi vào chi tiết, thiết kế từng biểu đồ sẽ được trình bày cùng với kết quả thực nghiệm ở Chương 5.

## CHƯƠNG 5. KẾT QUẢ THỰC NGHIỆM

Chương 5 sẽ trình bày về việc ghép nối các thành phần đã được xây dựng ở Chương 4 thành một hệ thống hoàn chỉnh. Từ đó, chúng ta sẽ quan sát và đánh giá kết quả thực nghiệm của hệ thống.

### 5.1 Tích hợp

Triển khai cài đặt cụm Hadoop với kiến trúc đã trình bày ở Chương 4, chúng ta có cấu hình cụm như hình 5.1: với hai Datanode, dung lượng lưu trữ xấp xỉ 400GB. Hadoop cung cấp cho chúng ta một giao diện Web (với cụm này là ở địa chỉ <http://10.5.92.76:9870>) để có thể xem các thông tin về cụm, Namenode, Datanode, Filesystem ...

#### Summary

Security is off.

Safemode is off.

4,533 files and directories, 4,468 blocks (4,468 replicated blocks, 0 erasure coded block groups) = 9,001 total filesystem object(s).

Heap Memory used 634.91 MB of 919.5 MB Heap Memory. Max Heap Memory is 3.48 GB.

Non Heap Memory used 85.68 MB of 87.59 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	391.72 GB
Configured Remote Capacity:	0 B
DFS Used:	93.67 GB (23.83%)
Non DFS Used:	3.21 GB
DFS Remaining:	188.96 GB (75.42%)
Block Pool Used:	93.67 GB (23.83%)
DataNodes usages% (Min/Median/Max/stdDev):	50.7% / 43.25% / 47.83% / 3.16%
<a href="#">Live Nodes</a>	2 (Decommissioned: 0, In Maintenance: 0)
<a href="#">Dead Nodes</a>	0 (Decommissioned: 0, In Maintenance: 0)
<a href="#">Decommissioning Nodes</a>	0
<a href="#">Entering Maintenance Nodes</a>	0
<a href="#">Total Datanode Volume Failures</a>	0 (0 B)
Number of Under-Replicated Blocks	4468
Number of Blocks Pending Deletion (including replicas)	0
Block Deletion Start Time	Mon Jun 13 15:25:46 +0700 2022
Last Checkpoint Time	Mon Jun 13 15:25:27 +0700 2022
Enabled Erasure Coding Policies	RS-6-3-1024k

**Hình 5.1:** Cấu hình cụm Hadoop

Dữ liệu nhật ký đọc báo của người dùng được chuyển qua định dạng parquet và lưu hàng ngày, mỗi thư mục tương ứng với dữ liệu trong một ngày, dữ liệu Mobile và PC được lưu ở những đường dẫn khác nhau trên HDFS (hình 5.2).

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities

Browse Directory

/data/pageview\_pc

Go!

Show

25

entries

Search:

<input type="checkbox"/>	<div><div></div></div> Permission	<div><div></div></div> Owner	<div><div></div></div> Group	<div><div></div></div> Size	<div><div></div></div> Last Modified	<div><div></div></div> Replication	<div><div></div></div> Block Size	<div><div></div></div> Name	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 13 15:36	<a href="#">0</a>	0 B	<a href="#">2022-06-12</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 14 10:14	<a href="#">0</a>	0 B	<a href="#">2022-06-13</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 15 10:13	<a href="#">0</a>	0 B	<a href="#">2022-06-14</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 16 10:12	<a href="#">0</a>	0 B	<a href="#">2022-06-15</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 17 10:12	<a href="#">0</a>	0 B	<a href="#">2022-06-16</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 18 10:12	<a href="#">0</a>	0 B	<a href="#">2022-06-17</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 19 10:11	<a href="#">0</a>	0 B	<a href="#">2022-06-18</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 20 10:10	<a href="#">0</a>	0 B	<a href="#">2022-06-19</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 21 10:12	<a href="#">0</a>	0 B	<a href="#">2022-06-20</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 22 10:11	<a href="#">0</a>	0 B	<a href="#">2022-06-21</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 23 10:13	<a href="#">0</a>	0 B	<a href="#">2022-06-22</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 24 10:11	<a href="#">0</a>	0 B	<a href="#">2022-06-23</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 25 10:12	<a href="#">0</a>	0 B	<a href="#">2022-06-24</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 26 10:10	<a href="#">0</a>	0 B	<a href="#">2022-06-25</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 27 10:12	<a href="#">0</a>	0 B	<a href="#">2022-06-26</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 28 10:14	<a href="#">0</a>	0 B	<a href="#">2022-06-27</a>	<div><div></div></div>
<input type="checkbox"/>	<a href="#">drwxrwxrwx</a>	<a href="#">airflow</a>	<a href="#">supergroup</a>	0 B	Jun 29 10:12	<a href="#">0</a>	0 B	<a href="#">2022-06-28</a>	<div><div></div></div>


Hình 5.2: Cấu trúc thư mục lưu trữ ở HDFS

Đây là mẫu số một bản ghi ở định dạng parquet (hình 5.3).

time	guid	gender	age	browser	os	location
2022-07-14 18:42:12	6193794772065480245	1	6	14	1	-1
2022-07-14 11:55:33	2145062592112613436	1	4	14	1	5
2022-07-14 11:07:32	6389727711984422063	2	5	14	1	4
2022-07-14 15:44:16	2774124581984364804	2	4	17	1	4
2022-07-14 20:13:47	7439213806457887253	2	5	14	10	5
2022-07-14 12:14:34	1450452051245549868	1	4	14	1	46
2022-07-14 12:18:09	6587358095245536834	2	5	14	10	4
2022-07-14 11:44:16	8995984951247111032	1	5	14	10	5
2022-07-14 16:46:19	2415807663741060985	1	4	14	10	4
2022-07-14 01:36:49	8727081849250988532	1	5	14	1	32
2022-07-14 17:06:51	9086086041953349105	2	7	14	1	-1
2022-07-14 13:20:33	4674447642883713710	1	5	10	2	50
2022-07-14 17:17:34	7657793854245597326	2	6	10	2	-1
2022-07-14 11:27:22	6270233583419213193	1	5	14	10	4
2022-07-14 15:35:15	2950347521245978853	1	6	14	10	5
2022-07-14 00:06:34	5020195005583814359	1	4	4	9	115
2022-07-14 12:26:15	4611279761457409780	2	3	14	1	15
2022-07-14 16:51:25	764924797520236149	1	4	14	1	-1
2022-07-14 09:34:30	4465450111934293604	1	5	14	10	53
2022-07-14 15:14:16	3635459751953304681	1	5	14	10	5

Hình 5.3: Mẫu một số bản ghi ở định dạng parquet

Giao diện Web cho trình quản lý tài nguyên YARN có thể truy cập bằng địa chỉ `http://10.5.92.76:8088`, cung cấp thông tin về tài nguyên tính toán của cụm (lượng tài nguyên tối đa được cấp phát, lượng tài nguyên đang sử dụng...) Đồng thời, YARN cũng lưu trữ lịch sử, trạng thái của các công việc được thực hiện dưới sự điều phối của mình. Hình 5.4 là lịch sử các công việc, hình 5.5 thể hiện trạng thái và các giai đoạn đã/đang thực hiện của một công việc Spark thực hiện phân tích dữ liệu lưu ở HDFS.



Cluster

About Nodes

Node Labels

Applications

NEW

NEW, SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

All Applications

Cluster Metrics

Apps Submitted

0

Apps Pending

0

Apps Running

26

Apps Completed

0

Containers Running

<memory:0 B, vCores:0>

Used Resources

<memory:8 GB, vCores:8>

Total Resources

Cluster Nodes Metrics

Active Nodes

0

Decommissioning Nodes

0

Decommissioned Nodes

0

Lost Nodes

0

Scheduler Metrics

Scheduler Type

Capacity Scheduler

Scheduling Resource Type

[memory-mb (unit-M), vcores]

Minimum Allocation

<memory:1024, vCores:1>

Maximum Allocation

<memory:8192, vCores:4>

Show 20 ▼ entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU Vcores	Alloc Mem
application_1655108782032_0020	hiephm	user-view-analysis	SPARK		default	0	Fri Jul 8 11:00:26 +0700 2022	Fri Jul 8 11:00:27 +0700 2022	Fri Jul 8 11:14:55 +0700 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A
application_1655108782032_0025	hiephm	user-view-analysis	SPARK		default	0	Thu Jul 7 11:00:22 +0700 2022	Thu Jul 7 11:00:22 +0700 2022	Thu Jul 7 11:14:02 +0700 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A
application_1655108782032_0024	hiephm	user-view-analysis	SPARK		default	0	Wed Jul 6 11:00:22 +0700 2022	Wed Jul 6 11:00:22 +0700 2022	Wed Jul 6 11:14:09 +0700 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A
application_1655108782032_0023	hiephm	user-view-analysis	SPARK		default	0	Tue Jul 5 11:00:29 +0700 2022	Tue Jul 5 11:00:29 +0700 2022	Tue Jul 5 11:14:58 +0700 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A
application_1655108782032_0022	hiephm	user-view-analysis	SPARK		default	0	Mon Jul 4 11:00:25 +0700 2022	Mon Jul 4 11:00:26 +0700 2022	Mon Jul 4 11:13:57 +0700 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A
application_1655108782032_0021	hiephm	user-view-analysis	SPARK		default	0	Sun Jul 3 11:00:32 +0700 2022	Sun Jul 3 11:00:33 +0700 2022	Sun Jul 3 11:14:51 +0700 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A
application_1655108782032_0020	hiephm	user-view-analysis	SPARK		default	0	Sat Jul 2 11:00:21 +0700 2022	Sat Jul 2 11:00:22 +0700 2022	Sat Jul 2 11:16:25 +0700 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A
application_1655108782032_0019	hiephm	user-view-analysis	SPARK		default	0	Fri Jul 1 11:00:24 +0700 2022	Fri Jul 1 11:00:25 +0700 2022	Fri Jul 1 11:14:05 +0700 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A
application_1655108782032_0018	hiephm	user-view-analysis	SPARK		default	0	Thu Jun 30 11:00:23 +0700 2022	Thu Jun 30 11:00:24 +0700 2022	Thu Jun 30 11:13:29 +0700 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A

**Hình 5.4:** Giao diện Web của trình quản lý tài nguyên YARN

### Spark Jobs <sup>(?)</sup>

User: hiephm  
Total Uptime: 3.2 min  
Scheduling Mode: FIFO  
Active Jobs: 2  
Completed Jobs: 5

Event Timeline

#### Active Jobs (1)

Page: 1

1 Pages. Jump to 1. Show 100 items in a page. Go

Job id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
5	collectAsList at Analyst.java:62 collectAsList at Analyst.java:62	2022/07/12 11:03:18 (kill)	2 s	0/2	0/23

Page: 1

1 Pages. Jump to 1. Show 100 items in a page. Go

#### Completed Jobs (5)

Page: 1

1 Pages. Jump to 1. Show 100 items in a page. Go

Job id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
4	collectAsList at Analyst.java:62 collectAsList at Analyst.java:62	2022/07/12 11:02:34	43 s	1/1	18/18
3	collectAsList at Analyst.java:57 collectAsList at Analyst.java:57	2022/07/12 11:02:32	0.9 s	1/1 (2 skipped)	1/1 (25 skipped)
2	collectAsList at Analyst.java:57 collectAsList at Analyst.java:57	2022/07/12 11:01:41	50 s	1/1 (1 skipped)	7/7 (18 skipped)
1	collectAsList at Analyst.java:57 collectAsList at Analyst.java:57	2022/07/12 11:00:57	44 s	1/1	18/18
0	parquet at Analyst.java:33 parquet at Analyst.java:33	2022/07/12 11:00:42	8 s	1/1	1/1

Page: 1

1 Pages. Jump to 1. Show 100 items in a page. Go

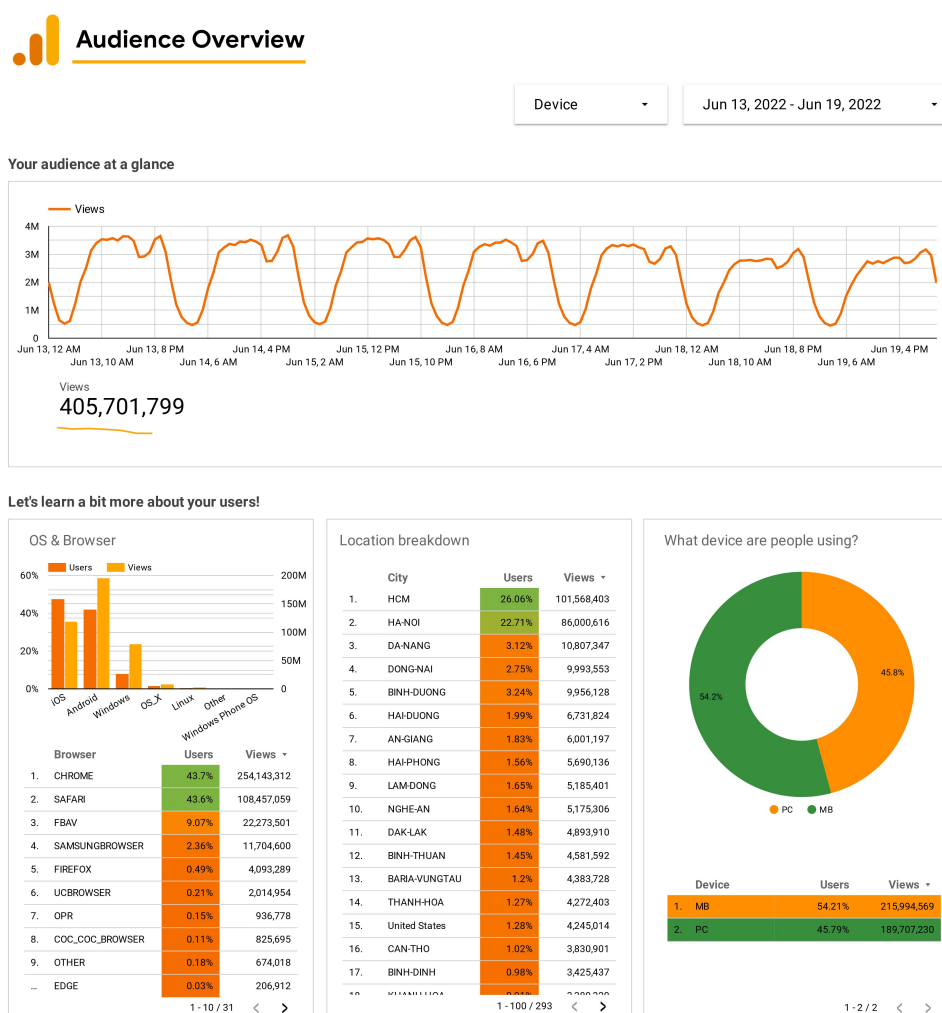
**Hình 5.5:** Trạng thái của một công việc Spark

## 5.2 Kết quả và đánh giá

Dữ liệu sau khi phân tích và lưu vào cơ sở dữ liệu sẽ được biểu diễn bằng các biểu đồ trực quan gọi là "Báo cáo phân tích". Báo cáo phân tích được thiết kế trên nền tảng Google Data Studio, giúp tạo ra các biểu đồ động, có thể thực hiện các thao tác như chọn khoảng thời gian, chọn thuộc tính, lọc ... Người dùng có thể tùy chỉnh biểu đồ để có các góc nhìn đa dạng hơn về dữ liệu từ đó đưa ra các kết luận chính xác hơn.

Giao diện báo cáo phân tích gồm có hai trang: **Tổng quan** và **Demographics**.

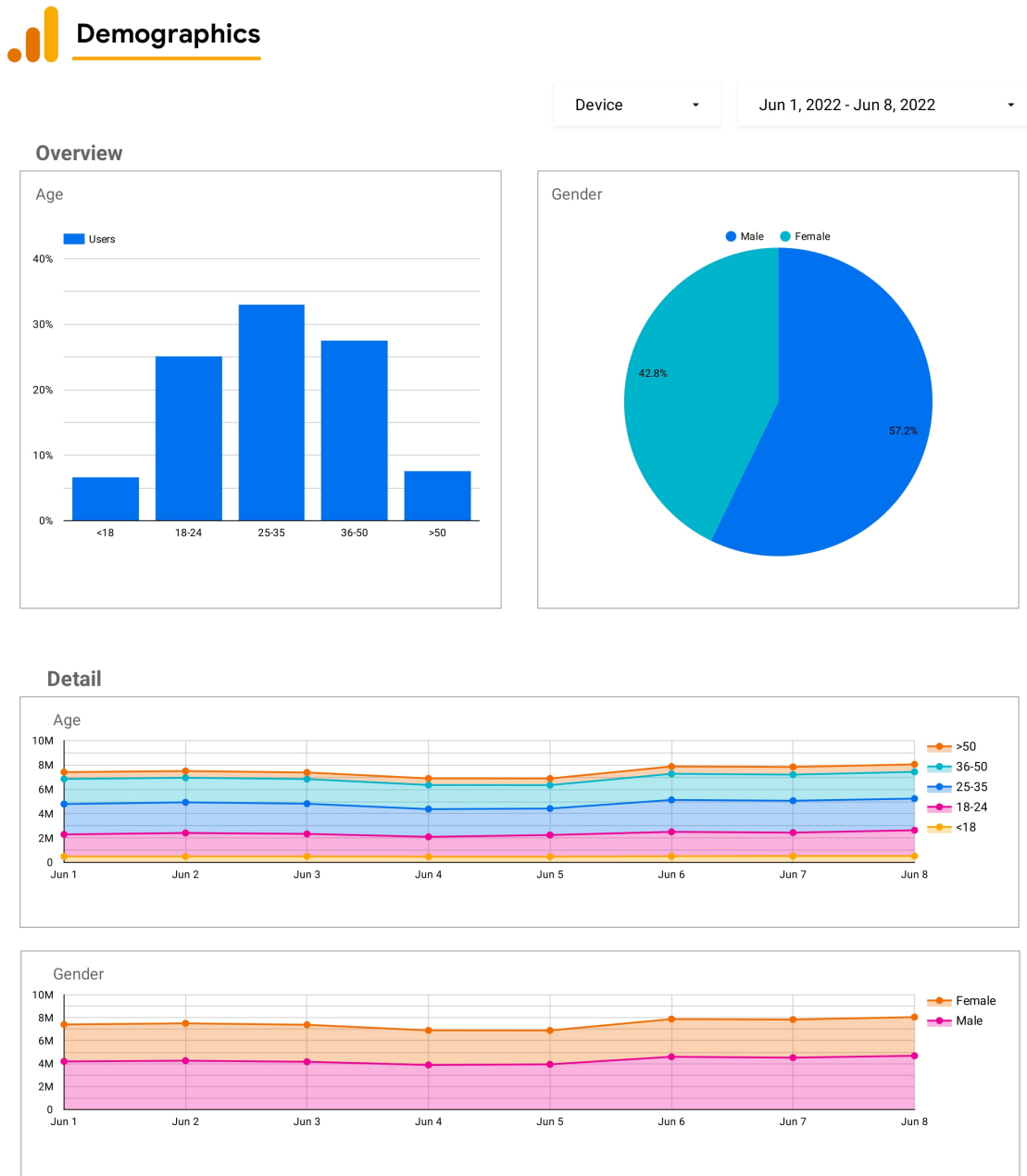
Trang tổng quan (hình 5.6) gồm các biểu đồ thể hiện số lượt xem theo thời gian và phân bố lượng người dùng theo các tiêu chí như loại thiết bị, vị trí địa lí... Cung cấp một góc nhìn tổng quát về tập hợp người đọc báo điện tử.



Hình 5.6: Giao diện tổng quan



Trang Demographics (hình 5.7) gồm các biểu đồ thể hiện sự phân bố của người dùng theo độ tuổi và giới tính, trong đó có cả biểu đồ theo thời gian.

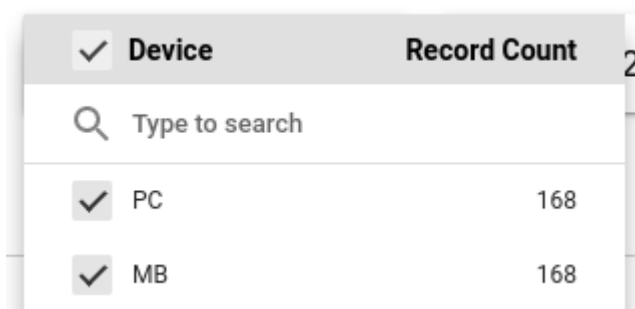


**Hình 5.7:** Giao diện Demographics

### 5.2.1 Chức năng chung

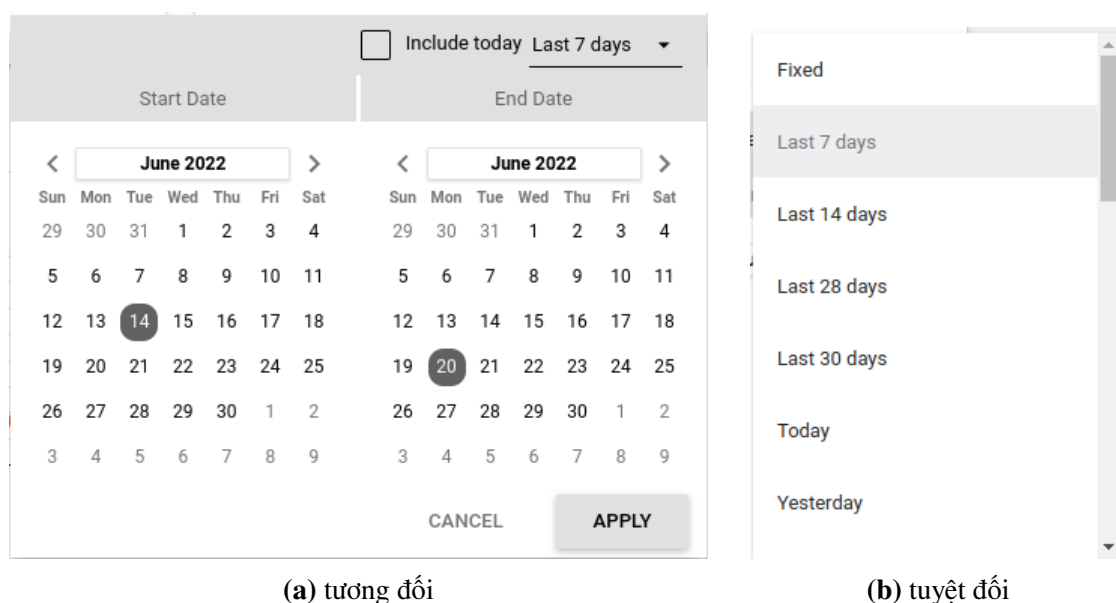
Đầu mỗi trang báo cáo đều có hai bộ lọc "Device" và "Time range".

Device (hình 5.8): lọc dữ liệu theo loại thiết bị được báo là máy tính (PC) và điện thoại (MB). Có thể chọn một trong hai loại thiết bị hoặc cả hai.



**Hình 5.8:** Bộ lọc Device

Time range (hình 5.9): lọc dữ liệu theo khoảng thời gian, có thể chọn khoảng thời gian tương đối (1 tuần, 1 tháng ... gần nhất) hoặc tuyệt đối (chọn cụ thể ngày bắt đầu và ngày kết thúc)

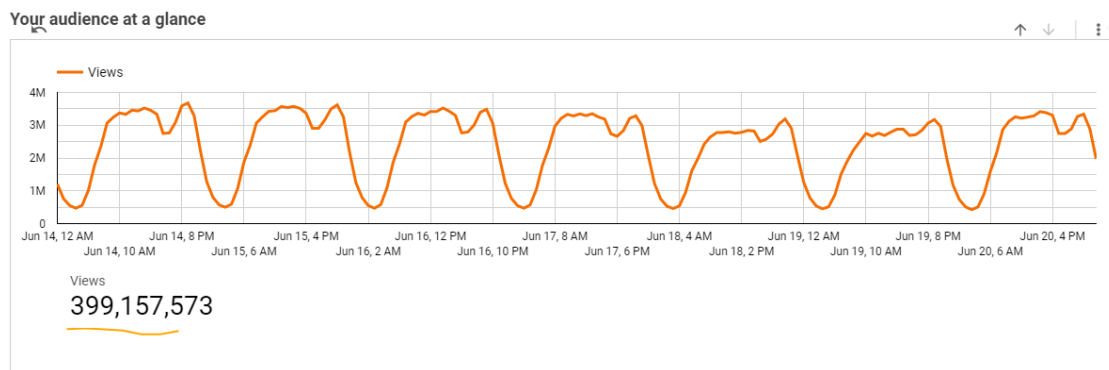


**Hình 5.9:** Bộ lọc khoảng thời gian

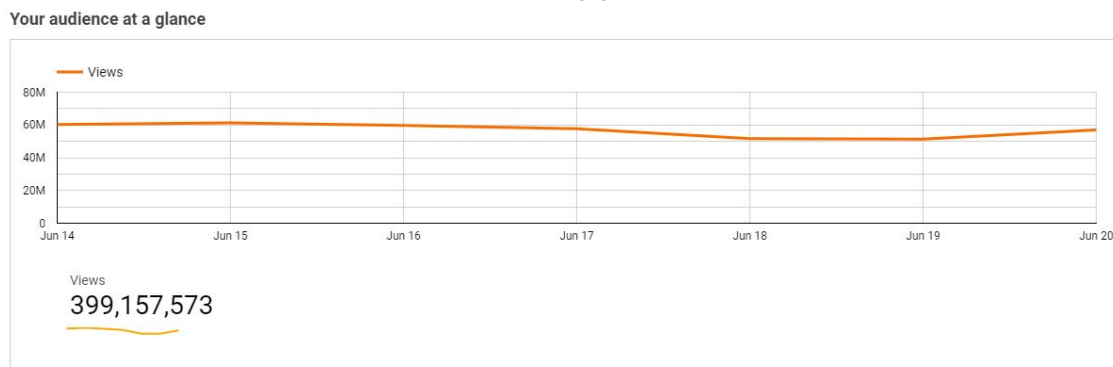
### 5.2.2 Các biểu đồ

**Biểu đồ lưu lượng truy cập theo thời gian:** hình 5.10.

Là biểu đồ dạng đường tạo thành từ các điểm dữ liệu với mỗi điểm biểu diễn số lượt xem tại một thời điểm. Theo mặc định số lượt xem sẽ được tính theo hàng giờ, người dùng có thể sử dụng phím mũi tên lên/xuống ở góc trên bên phải biểu đồ để đổi sang chế độ xem hàng ngày, tuần, tháng hoặc năm. Ở góc dưới bên trái là một biểu đồ phụ cho biết tổng lượt xem trong khoảng thời gian đang chọn.



(a) hàng giờ



(b) hàng ngày

**Hình 5.10:** Biểu đồ lưu lượng truy cập theo thời gian

Quan sát và so sánh biểu đồ ở chế độ hàng giờ và hàng ngày có thể thấy rằng lượng lượt xem hàng ngày không chênh lệch nhiều. Xét trong một ngày, khoảng thời gian từ đêm khuya đến sáng ngày hôm sau số lượt xem thấp hơn hẳn các khung giờ khác. Điều đó khá là dễ hiểu vì đó là lúc hầu hết mọi người đều nghỉ ngơi, đi ngủ. Xem biểu đồ hình 5.16 để có thể hiểu hơn về lưu lượng truy cập của người dùng trong ngày.

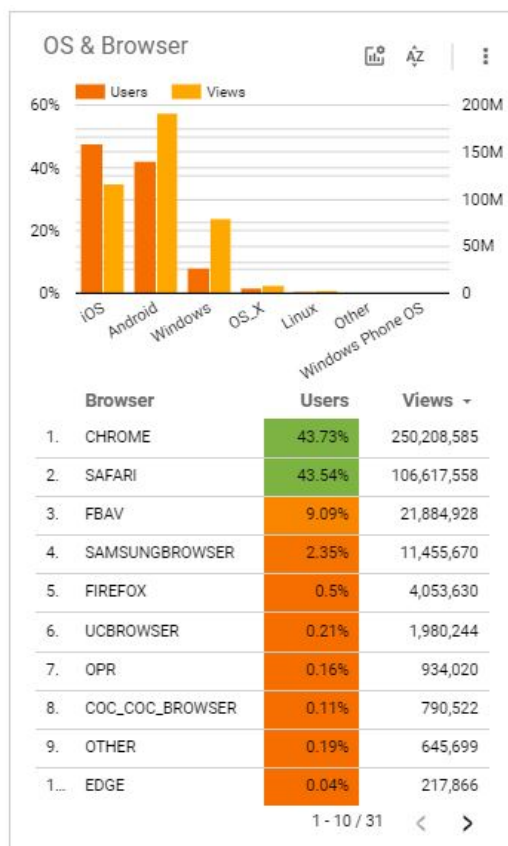
**Biểu đồ lưu lượng truy cập theo công nghệ sử dụng:** hình 5.11(a).

Gồm hai biểu đồ: (i) Hệ điều hành, là biểu đồ dạng cột, thể hiện sự phân bố người dùng hoặc số lượt xem của từng hệ điều hành. (ii) Trình duyệt, là biểu đồ dạng bảng gồm ba cột, tên trình duyệt - tỉ lệ người dùng - số lượt xem. Bảng được chia thành nhiều trang có thể sắp xếp theo thứ tự tăng dần hoặc giảm dần của số người dùng hoặc số lượt xem.

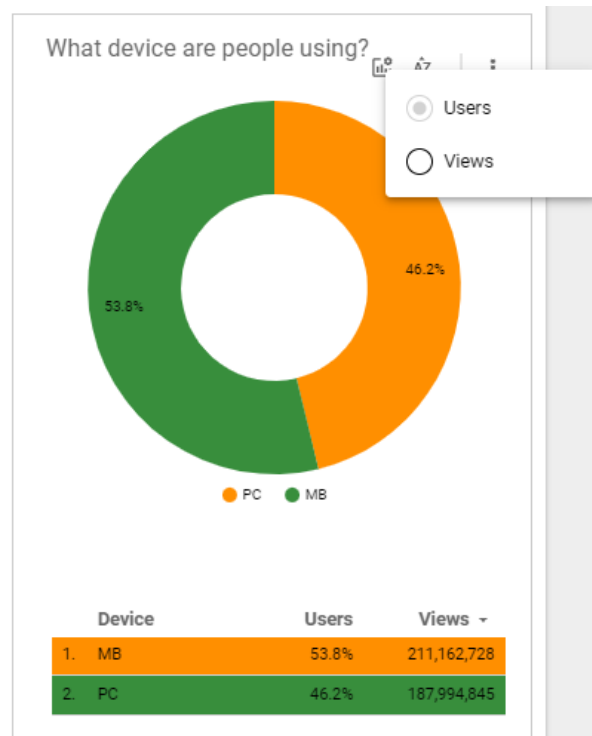
Với thiết bị điện thoại thì iOS và Android được sử dụng nhiều nhất, trên máy tính thì chủ yếu người dùng sử dụng Windows, OSX và Linux chiếm một lượng không đáng kể. Có rất nhiều và đa dạng các loại trình duyệt được sử dụng để đọc báo nhưng chủ yếu người dùng sử dụng Chrome và Safari.

**Biểu đồ lưu lượng truy cập theo loại thiết bị:** hình 5.11(b).

Là biểu đồ hình tròn thể hiện sự phân bố lượng người dùng/lượt xem theo loại thiết bị sử dụng người dùng sử dụng. Có hai loại thiết bị được định nghĩa là điện thoại (MB) và máy tính (PC). Xét theo lượng người dùng hay số lượt xem thì thiết bị điện thoại vẫn chiếm tỉ lệ lớn hơn(>50%).



(a) trình duyệt và hệ điều hành



(b) loại thiết bị

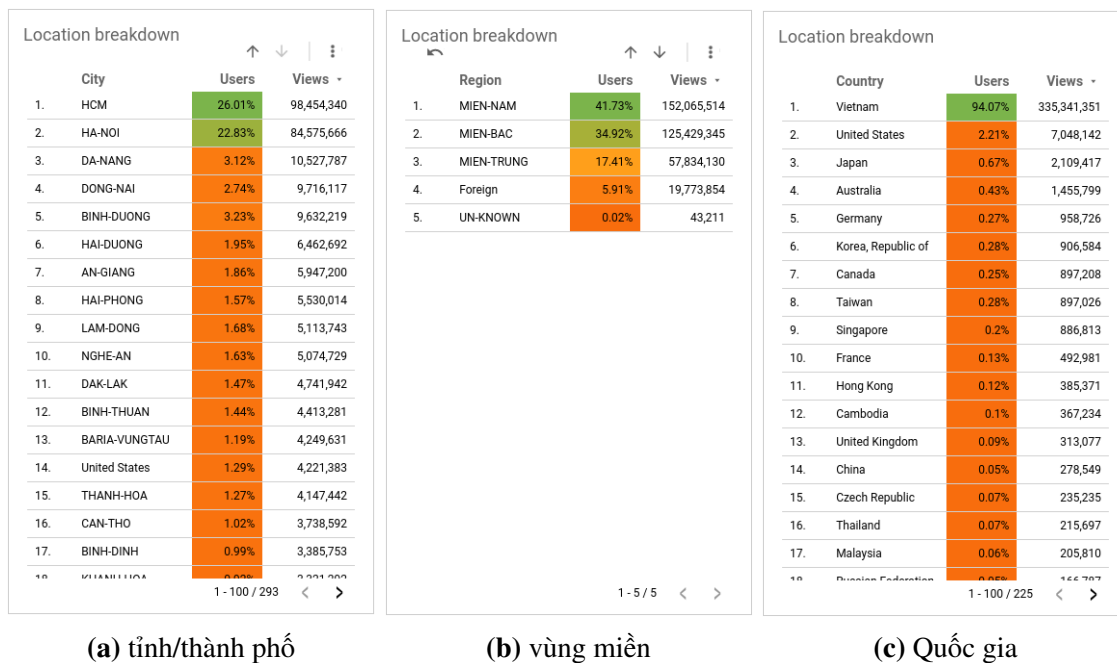
**Hình 5.11:** Biểu đồ lưu lượng truy cập theo thiết bị/công nghệ sử dụng

**Biểu đồ lưu lượng truy cập theo vị trí địa lý: hình 5.12.**

Là biểu đồ dạng bảng có ba cột (vị trí - tỉ lệ người dùng - số lượt xem), biểu diễn sự phân bố theo vị trí địa lý của người dùng.

Chế độ xem mặc định là phân bố theo tỉnh thành, để xem sự phân bố theo tỉnh - vùng miền - quốc gia, có thể sử dụng phím mũi tên lên/xuống ở góc trên bên phải biểu đồ để đổi giữa các chế độ (tương tự hình 5.10).

Bảng được chia thành nhiều trang có thể sắp xếp theo thứ tự tăng dần hoặc giảm dần của số người dùng hoặc số lượt xem.

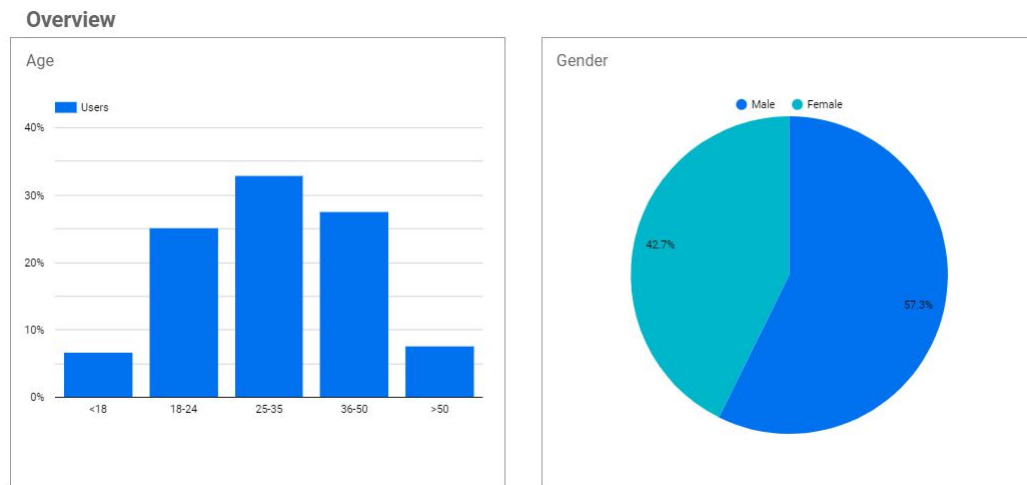


**Hình 5.12:** Biểu đồ lưu lượng truy cập theo vị trí địa lý

Quan sát biểu đồ ta thấy lượng người dùng trong nước chiếm 95%, lý do là dữ liệu đến từ các trang báo Tiếng Việt. Một lượng truy cập đến từ nước ngoài như Mỹ, Nhật, Úc cũng đáng kể, có thể chủ yếu đến từ người Việt tại nước ngoài.

Ở trong nước, sự phân bố người dùng/lượt xem phần nào đó tương đồng với sự phân bố dân cư khi Hà Nội và Thành phố Hồ Chí Minh dẫn đầu và chiếm phần lớn. Với vùng miền, miền Bắc và miền Nam chiếm tỉ lệ vượt trội so với miền Trung.

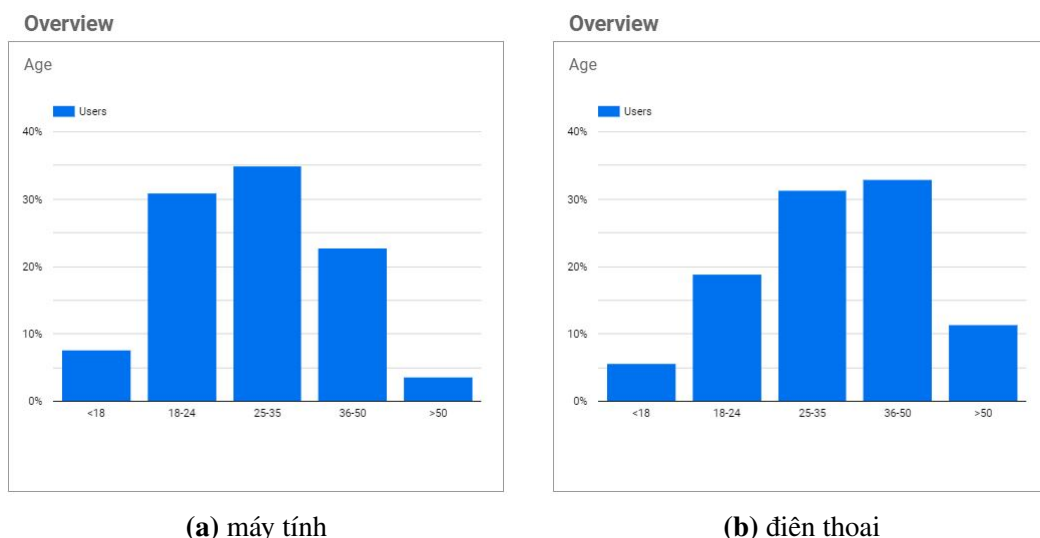
**Biểu đồ độ tuổi/giới tính của người dùng:** hình 5.13, thể hiện sự phân bố theo độ tuổi/giới tính của người dùng. Với độ tuổi, không xét cụ thể từng tuổi mà xét theo nhóm tuổi (có năm nhóm độ tuổi như ở biểu đồ dưới)



**Hình 5.13:** Tổng quan phân bố người dùng theo độ tuổi và giới tính

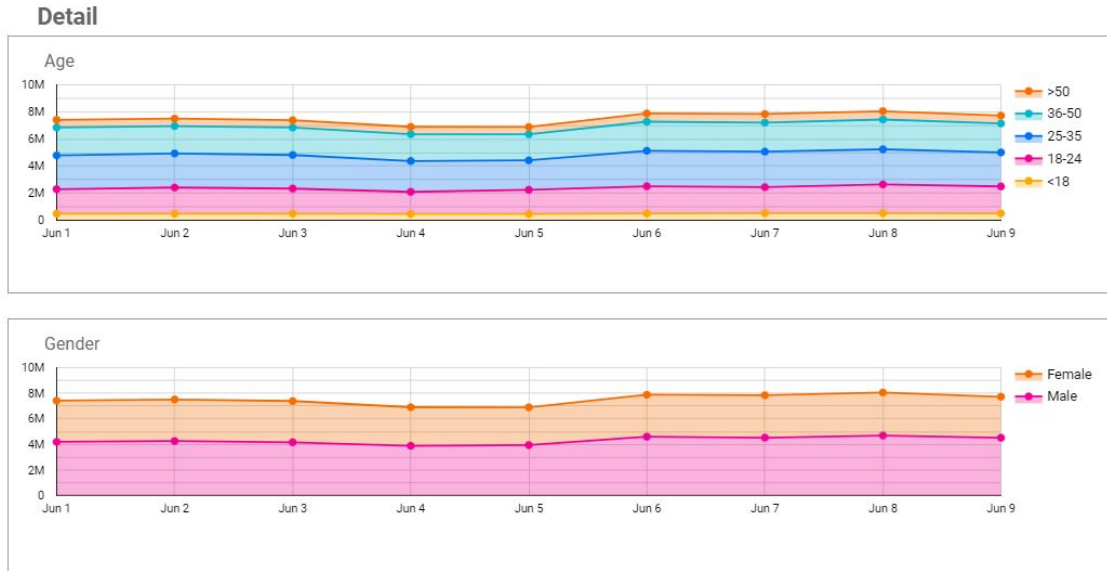
Nhìn biểu đồ ở trên (hình 5.13) có thể thấy rằng số lượng người đọc báo là nam giới lớn hơn nữ giới. Độ tuổi đọc báo chiếm phần lớn là trưởng thành và trung niên (từ 18-50 tuổi) khoảng 80%, 20% còn lại là người già trên 50 tuổi và người dưới 18 tuổi.

Một điều thú vị là khi quan sát độ tuổi của người dùng theo loại thiết bị thì sự phân bố đã có chút thay đổi. Với máy tính thì biểu đồ có xu hướng nghiêng về phía bên trái so với biểu đồ chung, và ngược lại, với điện thoại thì biểu đồ nghiêng về phía bên phải (hình 5.14). Có thể hiểu rằng, càng cao tuổi thì người đọc có xu hướng ít sử dụng máy tính để đọc báo hơn và thay vào đó là điện thoại.



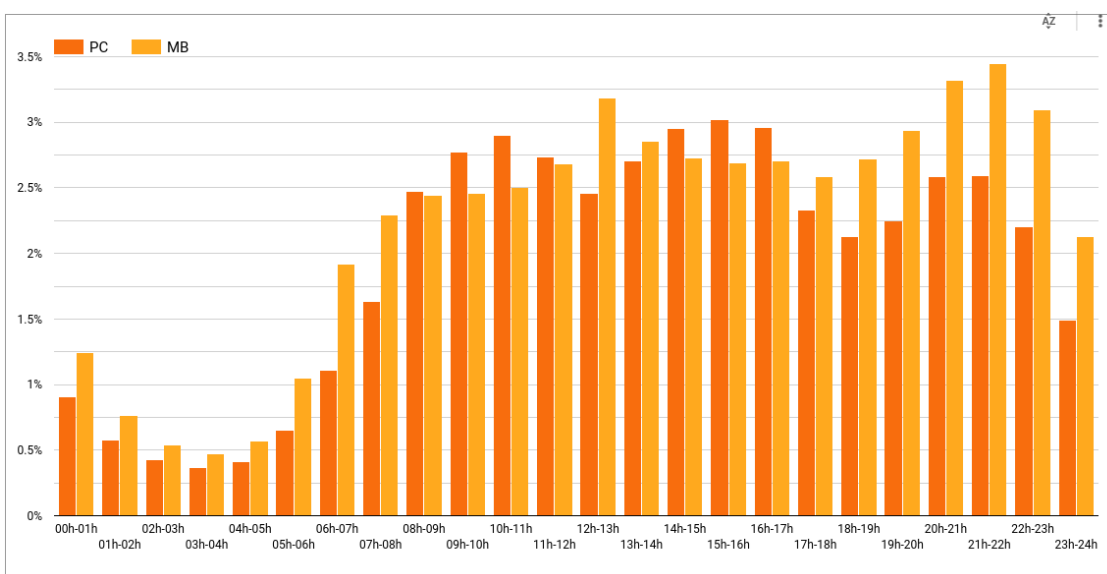
**Hình 5.14:** Phân bố người dùng theo độ tuổi khi lọc theo loại thiết bị

Ngoài ra có thể theo dõi cụ thể lượng người dùng theo độ tuổi và giới tính hàng ngày với biểu đồ ở hình 5.15. Nhìn biểu đồ có thể thấy rằng, cũng như số lượt xem hàng ngày, số lượng người dùng và sự phân bố theo độ tuổi/giới tính khá ổn định qua các ngày.



**Hình 5.15:** Lượng người dùng theo độ tuổi và giới tính hàng ngày

**Biểu đồ lưu lượng truy cập theo khung giờ:** hình 5.16, là biểu đồ cột, mỗi cột biểu diễn số lượt xem ở một khung giờ trong ngày, có hai loại cột tương ứng với hai loại thiết bị PC và MB.



**Hình 5.16:** Lưu lượng truy cập theo khung giờ trong ngày

Quan sát biểu đồ ta thấy khung giờ cao điểm đối với hai loại thiết bị là khác nhau, với thiết bị Mobile có hai khung giờ số lượt xem cao đó là từ 12h-13h và từ 20h-23h, với thiết bị PC thì lưu lượng truy cập trải đều từ 8h-17h (có thấp hơn ở 12h-13h). Giờ hành chính ở Việt Nam thường là 8h đến 17h và nghỉ trưa vào 12h-13h, kết hợp với dữ kiện ở trên có thể nhận xét rằng người dùng thường sử dụng máy tính để đọc báo trong giờ hành chính và trong giờ nghỉ (nghỉ trưa, đêm ...) họ thường sử dụng thiết bị di động.



## CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Đồ án đã xây dựng thành công một mô hình hệ thống để giải quyết bài toán phân tích dữ liệu người dùng báo điện tử. Thông tin từ kết quả đầu ra của hệ thống là báo cáo phân tích sẽ giúp doanh nghiệp vận hành các chiến dịch quảng cáo trên báo điện tử một cách tối ưu hơn. Mở rộng ra, hệ thống cũng có thể giải quyết được các bài toán liên quan đến phân tích dữ liệu lớn khác.

Hệ thống có các ưu điểm: (i) dễ dàng mở rộng quy mô, (ii) công cụ phân tích dữ liệu lớn mạnh mẽ, (iii) tạo báo cáo phân tích có mức độ tùy chỉnh/tương tác cao mà không cần quá nhiều kinh nghiệm và công sức xây dựng giao diện.

Trong quá trình làm đồ án, em đã học được:

- Quy trình phân tích và thiết kế một hệ thống để giải quyết một bài toán cụ thể.
- Kỹ năng cài đặt, sử dụng các công nghệ Hadoop và Spark để lưu trữ và tính toán trên dữ liệu lớn.
- Kỹ năng thiết kế cơ sở dữ liệu.
- Sử dụng thành thạo công cụ Google Data Studio để thiết kế, xây dựng báo cáo phân tích.

Với sự hướng dẫn tận tình của thầy Trần Việt Trung, em đã hoàn thành đồ án này dù còn một vài thiếu sót.

### 6.2 Hướng phát triển

Hiện tại, đồ án này mới chỉ khai thác một phần thông tin có được từ dữ liệu nhật ký truy cập báo điện tử của người dùng. Việc phân tích dữ liệu cũng đang được thực hiện cố định hàng ngày, có nghĩa là những dữ liệu như số người dùng sẽ không chính xác hoàn toàn khi tổng hợp theo các khoảng thời gian.

Vì vậy trong tương lai, em có một số định hướng phát triển đồ án như sau:

- Khai thác tối đa các thông tin có ích từ nhật ký truy cập báo điện tử của người dùng. Ví dụ: trang báo/chuyên mục người dùng truy cập, giúp hiểu về xu hướng/sở thích của tập người dùng, làm cho báo cáo phân tích trở nên đa dạng với nhiều thông tin hữu ích hơn.
- Tính toán dữ liệu định kỳ hàng tuần, tháng, quý để bổ sung thêm thông tin.
- Nghiên cứu triển khai mô-đun xử lý dữ liệu theo luồng sử dụng Kafka, Spark Streaming để số liệu trên báo cáo gần với thời gian thực.

Với những cải tiến đó, em hy vọng có thể hoàn thiện đồ án hơn, giúp tạo ra một mô hình hệ thống có ích trong việc phân tích dữ liệu lớn và được áp dụng để giải quyết các vấn đề thực tế trong môi trường doanh nghiệp.

## TÀI LIỆU THAM KHẢO

- [1] S. Sagirolu and D. Sinanc, “Big data: A review,” in *2013 international conference on collaboration technologies and systems (CTS)*, IEEE, 2013, pp. 42–47.
- [2] *Apache hadoop*. [Online]. Available: <https://hadoop.apache.org> (visited on 08/04/2022).
- [3] *Apache spark*. [Online]. Available: <https://spark.apache.org> (visited on 08/04/2022).
- [4] V. Dinh, *Hướng dẫn sử dụng google data studio cho người mới bắt đầu*. [Online]. Available: <https://mangoads.vn/learn/huong-dan-su-dung-google-data-studio/> (visited on 08/05/2022).