**ĐẠI HỌC QUỐC GIA TP.HCM**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH**

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**NGHIÊN CỨU VÀ XÂY DỰNG TẬP DỮ LIỆU HỮU ÍCH CHO BÀI TOÁN CDNS (CONTENT DELIVERY/DISTRIBUTION NETWORKS) DỰA TRÊN PHÂN TÍCH LOG FILES CỦA HỆ THỐNG SERVERS**

**HỘI ĐỒNG: HỆ THỐNG VÀ MẠNG**

**GVHD : PGS.TS THOẠI NAM**

**GVPB : ThS NGUYỄN CAO ĐẠT**

**SVTH : NGUYỄN TUẤN MINH (1412305)**

TP. HỒ CHÍ MINH, 06/2018

**LỜI CAM ĐOAN**

**LỜI CẢM ƠN**

**TÓM TẮT LUẬN VĂN**

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

Chương 1 : Tổng quan

1.1 Giới thiệu đề tài

1.2 Lí do chọn đề tài

1.3 Mục tiêu của đề tài

1.4 Đối tượng và phạm vi nghiên cứu

1.5 Phương pháp nghiên cứu và hướng tiếp cận

1.6 Cấu trúc luận văn

Chương 2: Kiến thức nền tảng

2.1 Hệ thống phân phối nội dung CDN

2.2 Cấu trúc log file

2.3 Apache Spark

2.4 Apache Zeppelin

2.5 Tableau

Chương 3 : Hiện thực

3.1 Cách thức phân tích Log file

3.2 Các trực quan hóa dữ liệu

Chương 4 : Đánh giá

4.1 Môi trường thực nghiệm

4.2 Kết quả Log file

4.3 Các biểu đồ và ý nghĩa

Chuong 5 : Tổng kết và hướng phát triển

DANH MỤC TỪ VIẾT TẮT

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

**CHƯƠNG 1 : TỔNG QUAN**

## 1.1 GIỚI THIỆU ĐỀ TÀI

Hiện nay chúng ta đang thấy sự bùng nổ mạnh mẽ của Internet và các nội dung trên mạng. Càng ngày càng phát triển có tác động đến việc các hoạt động của các ứng dụng và chất lượng dịch vụ của các hoạt động. Ngày càng tạo ra nhiều nội dung mới nên việc lưu lượng trên mạng được truy cập quá nhiều sẽ gây ra tắc nghẽn cho các lần truy cập. Để khắc phục sự cố này, các kỹ thuật giúp cải thiện việc truy cập và tránh các sự tắc nghẹn cho thấy rất quan trọng. Các kỹ thuật như: định cỡ mạng, QoS,.

Cũng như các cách trên thì kỹ thuật để giúp mạng hoạt động tốt hơn là giải pháp mạng phân phối nội dung (Content Delivery Networks-CDN) nhằm tránh sự tắc nghẽn của các hoạt động trên mạng. Các Client kết nối với Server nếu đi qua vùng bị tắc nghẽn sẽ bị trì trệ thay vào đó Client sẽ kết nối với các Server thay thế để thực hiện tốt hoạt động của mình. Như vậy sẽ trách được sự ùn tắc trong mạng và có thể khả năng tốc độ đường truyền sẽ cao hơn. Lưu trữ ở các Server thay thế như thế nào cũng là vấn đề được chú ý đến. Xây dựng các dữ liệu lưu trữ trên các Server thay thế được tính toán kĩ hơn. Chúng ta nên lưu lại những nội dung mà mọi người chú ý, được truy cập nhiều hơn sẽ giúp các hoạt động trên mạng tốt hơn. Phân bố nội dung trên mạng cho phù hợp với các Server để người dùng truy cập thông tin nhanh nhất mà ít bị tắc nghẽn. Để có thể tìm ra các nội dung nên được ở những nơi khác nhau. Ở các Server sẽ có lưu lại các hoạt động của các người dùng truy cập đến. Qua đó tìm hiểu cách thức phân tính và tìm ra các nội dung nên được lưu lại sau này. Từ việc muốn tìm ra nội dung trên, tìm hiểu và xây dựng tập dữ liệu cho việc phân bố nội dung ở các Server, cũng như thể hiện các ý nghĩa từ việc lưu lại các hoạt động của Client ở Server. Rõ ràng nếu lưu lại được các nội dung phù hợp ở các Server thay thế thì sẽ giảm bớt đi sự tắc nghẽn cho hoạt động trong mạng.

## 1.2 LÝ DO LỰA CHỌN ĐỀ TÀI

Cũng như được nêu ở trên, các nội dung ngày càng nhiều và việc truy cập cũng sẽ rất lớn. Nên giải pháp mạng phân bố nội dung trên mạng (CDN) là giải pháp tốt để tránh đi sự tắc nghẽn, mạng lại hiểu qủa đáp ứng yêu cầu người sử dụng. Những lợi ích mà giải pháp CDN mạng như khả năng phản ứng nhanh, giảm tắc nghẽn, đáp ứng yêu cầu người truy cập. CDN như một hệ thống mạng được người dùng, nhà cung cấp dịch vụ,...quan tâm, tìm hiểu và ứng dụng. Qua đó luận văn của tôi sẽ tìm hiểu và xây dựng các tập dữ liệu cho giải pháp CDN.

## 1.3 MỤC TIÊU ĐỀ TÀI

- Xây dựng tập dữ liệu thống kê số truy cập của các nội dung trên mạng

- Xây dựng tập dữ liệu về vị trí của các người dùng

- Phân tích và biểu thị các ý nghĩa từ các tập dữ liệu

## 1.4 ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Mục đích nghiên cứu của luận văn là xây dựng các tập dữ liệu và các nội dung cho mạng CDN. Qua đây sử dụng các file lưu trữ lại các hoạt động ở Server để phân tích. Sử dụng log file tại trường làm file thực thi cho bài luận văn. Log file của website trường cụ thể là của web: www.hcmut.edu.vn làm đối tượng của phạm vi nghiên cứu.

# 1.5 PHƯƠNG PHÁP NGHIÊN CỨU VÀ HƯỚNG TIẾP CẬN

Xây dựng tập dữ liệu và ý nghĩa trực quan từ log file ban đầu là một khăn trong lúc phân tích và xử lý. Log file là một khối dữ liệu rất lớn, việc phân tích, nó cũng gây tốn khá nhiều thời gian. Chúng ta cần hiểu về định dạng của log file sau đó chia nhỏ ra để loại bỏ đi bớt các phần mình không xử lý làm cho nó nhỏ lại.

Mỗi log file có kích thước rất lớn từ GB đến TB,... do đó việc sử dụng các ngôn ngữ, công cụ để phân tích sao cho phù hợp. Sử dụng các framework sẽ có thể nhanh hơn cho việc tạo ra dữ liệu mới có ý nghĩa nhiều hơn.

## 1.6 CẤU TRÚC LUẬN VĂN

Luận văn được chia thành 5 chương, nội dung của mỗi chương như sau:

Chương 1: Tổng quan

Chương đầu giới thiệu ngắn gọn về sự phát triển của đề tài. Bên cạnh đó, chương này tôi cũng nêu về lý do, mục tiêu, đối tượng và phạm vi nghiên cứu đề tài. Giới thiệu về cách thức nghiên cứu và hướng tiếp cận thực hiện cho đề tài.

Chương 2 : Kiến thức nền tảng

Chương này cung cấp các kiến thức nền tảng để thực hiện ở phần sau của luận văn. Các phần của chương này nói về các hệ thống phân nối nội dung(CDN), cấu trúc của log file ở trường cụ thể hơn là access log, cũng có thêm các công cụ hỗ trợ thực hiện như : Apache Spark, Zeppelin, Tableau.

Chương 3 : Hiện thực

Chương 4: Đánh giá

Chương 5: Tổng kết và hướng phát triển

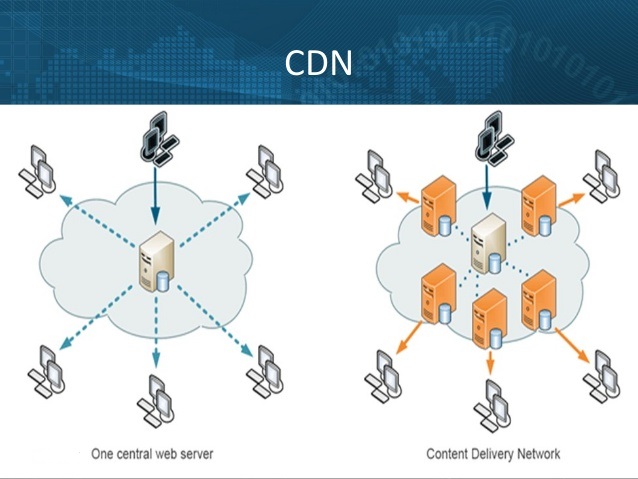
# CHƯƠNG 2 : KIẾN THỨC NỀN TẢNG

Chương này giới thiệu sơ bộ về các kiến thức làm nên tảng phục vụ cho việc thực hiện các thành phần ở phía sau. Phần đầu tiên giới thiệu về hệ thống mạng phân phối nội dung (CDN). Kế tiếp nói về nội dung log file sẽ đem phân tích cụ thể về các thành phần. Ở các phần tiếp theo sẽ giới thiệu về các công cụ cho phân tích đề tài: Apache Spark , Zeppelin, Tableau.

## 2.1 Hệ thống phân phối nội dung (CDN)

Nhằm cải thiện hoạt động của các người sử dụng, tránh đi hiện tượng “thắt cổ chai” khi các Client kết nối đến các Server nên thực hiện giải pháp mạng phân phối nội dung (CDN). Mạng phân phối nội dung(CDN) là mạng phân phối địa lý của máy chủ và các trạm thay thế. Mục đích là phân phối dịch vụ không gian liên quan đến người dùng cuối để cung cấp tính khả dụng cao và hiệu suất cao. CDN phục vụ một phần lớn nội dung Internet ngày nay, bao gồm các đối tượng web (văn bản, đồ họa và tập lệnh), các đối tượng tải xuống (tệp phương tiện, phần mềm, tài liệu), ứng dụng (thương mại điện tử, cổng), phương tiện phát trực tiếp, phát trực tuyến theo yêu cầu phương tiện truyền thông và mạng xã hội. Các nhà khai thác CDN phân phối các nội dung của các nhà chủ sở hữu nội dung cho các người dùng của họ.

Các nút CDN thường được triển khai ở nhiều vị trí, thường xuyên qua nhiều xương sống. Lợi ích bao gồm giảm chi phí băng thông, cải thiện thời gian tải trang hoặc tăng tính sẵn có của nội dung toàn cầu. Số lượng các nút và máy chủ tạo nên một CDN khác nhau, tùy thuộc vào kiến ​​trúc, một số đạt đến hàng ngàn nút với hàng chục nghìn máy chủ.



Hình 2.1 hình ảnh về mạng phân phối nội dung CDN

// <https://www.slideshare.net/saifmtter/cdn-52462637>

Các yêu cầu về nội dung thường được định hướng bởi các thuật toán cho các nút tối ưu theo một cách nào đó. Khi tính toán khoảng cách giữa các nút với nhau, giữa các nút với server gốc. Tính toán các nội dung phù hợp để lưu lại ở các nút một cách hiệu quả nhất. Các mạng lưới CDN có quy mô từ lớn đến nhỏ khác nhau. Thực hiện trên các châu lục, các quốc gia với quốc gia, các thành phố với nhau,…

Cách hoạt động của mạng CDN:

Thay vì các người dùng truy cập đến nội dung họ mong muốn được lưu trực tiếp trên Server gốc thì người dùng có thể sẽ truy cập đến các nút có chứa nội dung đó ở gần nhất có thể. Khi các nội dung mới được truy cập thì các nút sẽ lưu lại các nội dung mới để có thể xuất ra thông tin nhanh nhất cho người dùng có truy cập đến. Phân phối theo vị trí địa lý, các nội dung ở từng khu vực ở lưu lại cho hợp lý.

/// thêm chi tiết

Các lợi ích của hệ thống mạng phân phối nội dung:

+ Lợi ích cho các nhà thực hiện CDN:

- Giảm tải cho hệ thống máy chủ vận hành chính

- Các file tĩnh của website sẽ được bố trí trên các cụm máy chủ CDN Network giúp cho các máy chủ giảm tải trong quá trình vận hành hệ thống

- Tốc độ truy cập vào website nhanh hơn dù ở bất kỳ nơi đâu

- Dùng cơ chế xác định vị trí máy chủ gần nhất so với client giúp cho việc truyền tải dữ liệu nhanh hơn giúp website bạn có tốc độ truy xuất nhanh hơn dù ở bất kỳ nơi đâu

- Tương thích với các mã nguồn thông dụng “wordpress, joomla, drupal, magento”

- Tiết kiệm chi phí đầu tư nâng cấp cho hệ thống máy chủ hiện tại

- Thay vì phải trang bị nhiều máy chủ đặt tại nhiều nơi bạn có thể dùng dịch vụ CDN để tiết kiệm chi phí đầu tư thiết bị và tập trung vào công việc kinh doanh của bạn và mang lại hiệu quả cao hơn

- Giúp tăng thêm đối tượng truy cập ở nhiều nơi trên thế giới

- Cùng với việc mở rộng phạm vi truy cập sẽ giúp bạn tìm kiếm được các khách hàng tiềm năng và mở rộng hoạt động kinh doanh của bạn sang các khu vực và quốc gia khác

- Giúp tăng thêm đối tượng truy cập ở nhiều nơi

- Chỉ phải trả tiền theo lưu lượng băng thông đã sử dụng

+ Lợi ích cho các người dùng:

- Tiết kiệm băng thông đáng kể đối với các dữ liệu tĩnh (hình ảnh, css, javascript)

- Tăng tốc độ truy cập website, load nội dung nhanh, giảm thiểu độ trễ, giật hình khi truy cập và xem các trang website phân phối nội dung như: Movies, Video clip, TVC, vvv…

- Cho phép người dùng Internet có thể tương tác nhanh chóng, gia tăng sự hài lòng khi tiếp cận website trong thời gian thực

+ Các đối tượng cần dùng CDN :

- Các Website có lượng truy cập lớn, Website chứa nhiều nội dung tĩnh (hình ảnh, css, javascript). Sử dụng CDN sẽ tiết kiệm hơn là dùng Server riêng cho các website tầm trung

- Máy chủ gốc đặt ở xa đối tượng người dùng hoặc cần phân phối nội dung với chất lượng tốt nhất trên toàn thế giới

- Các nhà cung cấp dịch vụ Media, các doanh nghiệp, cá nhân sử dụng hạ tầng CDN để phân phối nội dung ( Movies, Video clip, TVC…) trên Internet nhằm quảng bá và kinh doanh các sản phẩm dịch vụ do doanh nghiệp, cá nhân cung cấp tới người dùng cuối

- Đối với các đài truyền hình, đơn vị có thể phát triển kênh truyền hình cung cấp cho người xem thông qua mạng Internet trên trang web của chính đài truyền hình

## 2.2 Cấu trúc log file

Hoạt động của các công ty, trường học và các server khác xảy ra liên tục ngày này qua ngày khác. Tại các server luôn phải ghi lại các hoạt động xảy ra ở server của mình. Những hành động tác động đến nội dung, truy cập đến trang web,các lỗi xảy ra,... sẽ được server nhận biết để lưu trữ lại trong server. Các file chứa các nội dung như vậy là gọi là log. Có nhiều loại log ở từng server khác nhau như: access log, error log, sql log, system log,...

Mỗi loại log sẽ có các cấu trúc khác tùy vào người cấu hình bên server. Dưới đây là một cấu trúc mẫu của access log :

113.162.189.51 - - [08/Nov/2016:12:00:29 +0700] "GET /includes/js/jquery/jquery-1.6.4.min.js HTTP/1.1" 200 91669 "http://www.hcmut.edu.vn/vi" "Mozilla/5.0 (Linux; Android 5.1.1; D2502 Build/19.4.A.0.182; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/46.0.2490.76 Mobile Safari/537.36"

Gồm các thành phần :

+ Địa chỉ IP (Ip Address): 113.162.189.51: đây là một địa chỉ ip public của người dùng truy cập đến trang web

+ Tên truy cập (Username) : - - : sẽ có liên quan khi truy cập vào các trang web có bảo vệ bằng mật khẩu

+ Thời gian (Timestamp): [08/Nov/2016:12:00:29 +0700]: trường này chứa thời gian truy cập và timezone của web server

+ Yêu cầu truy cập (Accsess request) : "GET /includes/js/jquery/jquery-1.6.4.min.js HTTP/1.1" : sử dụng GET request là hiển thị trang web cho file "/includes/js/jquery/jquery-1.6.4.min.js" bằng giao thức "HTTP/1.1"

+ Mã trạng thái kết quả (Result status code) : 200 : mã trạng thái kết qủa 200 là thành công. Nếu đường dẫn không tồn tại thì mã là 404. Còn nhiều mã lỗi khác nhau.

+ Dung lượng chuyển đi (bytes transferred) : 91669 là dung lượng byte chuyển đi.

+ Đường dẫn tham khảo (Referrer URL ): http://www.hcmut.edu.vn/vi : đây là trang web mà khách truy cập đến. Nó có thể là đường dẫn có liên kết từ các trang web khác hoặc nó chỉ là địa chỉ mà họ gõ vào để truy cập. Có thể là dấu "-" vì :

* Nó được nhập trực tiếp vào browser hoặc sử dụng bookmark
* Có thể là đường link dẫn từ bên ngoài ví dụ như link trong mail hay điện thoại
* Có thể chỉnh browser của họ không gửi link tham khảo

+ User agent : "Mozilla/5.0 (Linux; Android 5.1.1; D2502 Build/19.4.A.0.182; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/46.0.2490.76 Mobile Safari/537.36"

Giới thiệu về các browser dùng để truy cập và một số phiên bản

## 2.3 Apache Spark

Mọi thứ ngày càng phát triển nên việc sử dụng dữ liệu của mọi người ngày càng lớn. Càng tạo nhiều thêm nhiều dữ liệu mới để phục vụ cho cuộc sống. Nên việc lưu trữ và xử lý dữ liệu cũng là vấn đề được nhiều công ty chú ý đến. Xử lý các dữ liệu có dung lượng lớn sẽ chậm và tốn nhiều thời gian. Trong đó có những giải đã được đưa ra để phục vụ cho việc khai thác và xử lý dữ liệu như Apache Spark.

Apache Spark là một hệ thống mã nguồn mở cho phép thực hiện tính toán trên cụm nhằm tạo ra khả năng phân tích dữ liệu nhanh với 2 tiêu chí: nhanh về cả lúc chạy và nhanh cả lúc ghi dữ liệu.

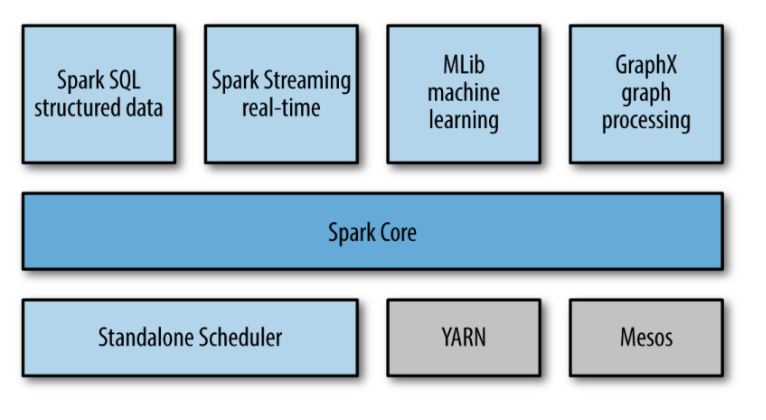
Spark được phát triển sơ khởi vào năm 2009 bởi AMPLab tại đại học California. Sau này, Spark đã được trao cho Apache Software Foundation vào năm 2013 và được phát triển cho đến nay

Để chạy chương trình nhanh hơn, Spark cung cấp một mô hình thực thi cho phép tối ưu các tính toán đồ thị một cách tùy ý (optimize arbitrary operator graphs) và hỗ trợ tính toán tại bộ nhớ trong giúp việc query dữ liệu nhanh hơn công nghệ tính toán dựa trên bộ nhớ ngoài (disk-based) như là Hadoop.

Để việc lập trình được đơn giản, Spark cung cấp một bộ API cho các ngôn ngữ Scala, Java và Python. Ngoài ra, chúng ta có thể tương tác với Spark bằng việc sử dụng Scala và Python shell để query dữ liệu tiện lợi hơn.

Spark ban đầu được phát triển cho hai loại thuật toán: thuật toán lặp (iterative algorithms) thường được sử dụng trong học máy và thuật toán tương tác data mining. Trong cả hai trường hợp, Spark có thể chạy nhanh gấp 100 lần so với Hadoop MapReduce. Tuy nhiên, chúng ta còn có thể sử dụng Spark cho các bài toán xử lý dữ liệu thông thường.

+ Các thành phần của Apache Spark



Hình 2.3 Hình ảnh thành phần của Apache Spark

Mọi ứng dụng Spark đều bao gồm một chương trình điều khiển chạy chức năng chính của người dùng và thực hiện các hoạt động song song khác nhau trên một cụm. Spark trừu tượng chính cung cấp là một tập dữ liệu được phân phối đàn hồi (resilient distributed dataset- RDD), là tập hợp các phần tử được phân đoạn trên các nút của cụm có thể hoạt động song song. RDD được tạo bằng cách bắt đầu với một tệp trong hệ thống tệp Hadoop (hoặc bất kỳ hệ thống tệp Hadoop nào khác được hỗ trợ) hoặc bộ sưu tập Scala hiện có trong chương trình trình điều khiển và chuyển đổi nó. Người dùng cũng có thể yêu cầu Spark duy trì RDD trong bộ nhớ, cho phép nó được tái sử dụng hiệu quả trên các hoạt động song song. Cuối cùng, RDD tự động phục hồi từ các lỗi của nút.

+ Spark SQL là mô-đun của Apache Spark để làm việc với dữ liệu có cấu trúc. Spark SQL cho phép truy vấn dữ liệu cấu trúc qua các câu lệnh SQL. Spark SQL có thể thao tác với nhiều nguồn dữ liệu như Hive tables, Parquet, và JSON.

Các tính năng của Spark SQL :

Tích hợp(Integrated) - Kết hợp liền mạch các truy vấn SQL với các chương trình Spark. Spark SQL cho phép bạn truy vấn dữ liệu có cấu trúc dưới dạng bộ dữ liệu phân tán (RDD) trong Spark, với các API được tích hợp trong Python, Scala và Java. Việc tích hợp chặt chẽ này giúp dễ dàng chạy các truy vấn SQL cùng với các thuật toán phân tích phức tạp.

Truy cập dữ liệu thống nhất(Unified Data Access) - Tải và truy vấn dữ liệu từ nhiều nguồn khác nhau. Lược đồ-RDD cung cấp một giao diện duy nhất để làm việc hiệu quả với dữ liệu có cấu trúc, bao gồm các bảng Apache Hive và các tệp JSON.

Khả năng tương thích Hive(Hive Compatibility)- Chạy truy vấn Hive chưa sửa đổi trên kho hiện có. Spark SQL tái sử dụng lối vào Hive và MetaStore, cho bạn khả năng tương thích đầy đủ với dữ liệu Hive, truy vấn và UDF hiện có. Đơn giản chỉ cần cài đặt nó cùng với Hive.

Kết nối chuẩn - Kết nối thông qua JDBC hoặc ODBC. Spark SQL bao gồm một chế độ máy chủ với kết nối JDBC và ODBC chuẩn công nghiệp.

// <https://www.tutorialspoint.com/spark_sql/spark_sql_introduction.htm>

+ Spark Streaming là phần mở rộng của API Spark core cho phép xử lý luồng có khả năng mở rộng, thông lượng cao, xử lý luồng có khả năng chịu lỗi của luồng dữ liệu trực tiếp. Dữ liệu có thể được nhập từ nhiều nguồn như Kafka, Flume, Kinesis hoặc TCP socket, và có thể được xử lý bằng cách sử dụng các thuật toán phức tạp được thể hiện với các hàm mức cao như map, reduce, join và window. Cuối cùng, dữ liệu đã xử lý có thể được đẩy ra ngoài hệ thống thành tệp, cơ sở dữ liệu và trang tổng quan trực tiếp. Trên thực tế, bạn có thể áp dụng các thuật toán xử lý đồ họa và học máy của Spark trên luồng dữ liệu.

+ Mlib machine learning

MLlib chứa nhiều thuật toán và tiện ích.

Một số thuật toán Machine Learning bao gồm:   
Phân loại: hồi quy logistic, naive Bayes, ...  
Hồi quy: hồi quy tuyến tính tổng quát, hồi quy sống còn(survival regression), ...  
Cây quyết định, cây có độ dốc

Các tập hợp thường xuyên, quy tắc kết hợp và khai thác mẫu tuần tự  
Các tiện ích dòng công việc ML bao gồm:  
Chuyển đổi tính năng: chuẩn hóa, chuẩn hóa, băm, ...  
ML Pipeline xây dựng  
Đánh giá mô hình và điều chỉnh siêu tham số  
ML persistence: lưu và tải các mô hình và đường ống  
Các tiện ích khác bao gồm:  
 Đại số tuyến tính phân tán: SVD, PCA, ...  
 Thống kê: thống kê tóm tắt, kiểm tra giả thuyết, ... b là thư viện học máy có khả năng mở rộng của Apache Spark.

+ GraphX graph processing

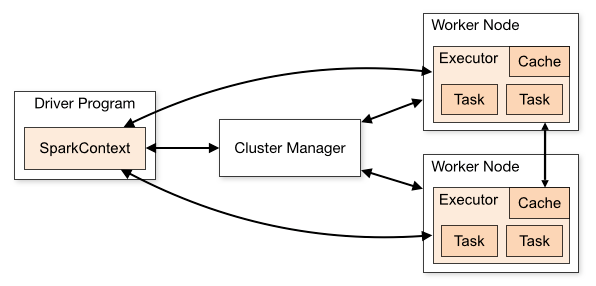
Việc phân tích dữ liệu lớn của Apache Spark ngày càng rộng và trở nên dễ dàng hơn, Spark triển khai các thuật toán hữu ích cho việc khai phá dữ liệu, phân tích dữ liệu, học máy, các thuật toán trên đồ thị. Sử dụng khả năng chạy trên cụm phân tán, Spark giải quyết các vấn đề như khả năng chịu lỗi và cung cấp API đơn giản để thực hiện tính toán song song.

GraphX là API của Apache Spark cho đồ thị và tính toán song song đồ thị. GraphX đơn giản hóa các nhiệm vụ phân tích biểu đồ. Có rất nhiều thuật toán trong lý thuyết đồ thì được sử dụng trong phân tích dữ liệu.

+Spark Core

Spark Core là nền tảng của toàn bộ dự án. Nó cung cấp các chức năng phân phối nhiện vụ, lịch trình và các chức năng nhập xuất cơ bản, được hiển thị thông qua giao diện lập trình ứng dụng. Mô hình lập trình chức năng hay bậc cao: các “Driver” gọi các hoạt động song song như map, filter, reduce trên RDD bằng cách chuyển một hàm tới Spark sau đó lên lịch thực hiện hàm song song trên các cụm. Các hoạt động kết nối với nhau từ RDD đầu tạo ra các RDD mới là không thay đổi và tính toán lười. RDD có thể thực hiện trên các đối tượng như Python, Scala, Java.

Spark có thể chạy trên nhiều loại Cluster Managers như Hadoop YARN, Apache Mesos hoặc trên chính cluster manager được cung cấp bởi Spark được gọi là Standalone Scheduler.

Cấu trúc spark

// https://spark.apache.org/docs/latest/cluster-overview.html

Hình 2.3.1 Cấu trúc thành phần của Apache Spark

Các ứng dụng Spark chạy như các bộ quy trình độc lập trên một cụm, được phối hợp bởi đối tượng SparkContext trong chương trình chính của bạn (được gọi là chương trình trình điều khiển – driver program).

Cụ thể, để chạy trên một cụm, SparkContext có thể kết nối với một số loại trình quản lý cụm(cluster manager) (hoặc trình quản lý cụm độc lập của Spark, Mesos hoặc YARN), phân bổ tài nguyên trên các ứng dụng. Khi đã kết nối, Spark sẽ thu nhận các thực thi trên các nút trong cụm, đó là các quá trình chạy các 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 trình thực thi. Cuối cùng, SparkContext gửi nhiệm vụ đến các trình thực thi để chạy.

Mỗi ứng dụng sẽ có các quy trình thực thi riêng của nó, nó sẽ ở lại trong suốt thời gian của toàn bộ ứng dụng và chạy các tác vụ trong nhiều luồng. Điều này có lợi ích của việc tách biệt các ứng dụng với nhau, trên cả hai mặt lịch trình (mỗi trình điều khiển lên lịch các nhiệm vụ riêng của mình) và bên thi hành (các nhiệm vụ từ các ứng dụng khác nhau chạy trong các JVM khác nhau).

Spark là bất khả tri đối với người quản lý cụm bên dưới. Miễn là nó có thể có được các quy trình thực thi, và các giao tiếp này với nhau, nó tương đối dễ dàng để chạy nó ngay cả trên một trình quản lý cụm cũng hỗ trợ các ứng dụng khác (ví dụ: Mesos / YARN)

Hệ thống hiện hỗ trợ ba trình quản lý cụm:

     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 có thể chạy các ứng dụng dịch vụ và MapReduce của Hadoop.  
     Hadoop YARN - trình quản lý tài nguyên trong Hadoop 2.  
     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 container.

## 2.4 Apache Zeppelin



Hình 2.4.1 Logo của Apache Zeppelin

Apache Zeppelin là web-based notebook dựa trên web cho phép phân tích dữ liệu, tương tác với các hình ảnh được tích hợp sẵn. Nó hỗ trợ nhiều ngôn ngữ với một khung thông dịch. Hiện tại, Zeppelin hỗ trợ các interpreter như Spark, Markdown, Shell, Hive, Phoenix, Tajo, Flink, Ignite, Lens, HBase, Cassandra, Elasticsearch, Geode, PostgreSQL và Hawq. Nó có thể được sử dụng để nhập dữ liệu, khám phá, phân tích dữ liệu và trực quan hóa bằng cách sử dụng notebooks tương tự IPython Notebooks.

Đặc biệt, Apache Zeppelin cung cấp tích hợp Apache Spark tích hợp. Bạn không cần phải xây dựng một module, plugin hoặc thư viện riêng cho nó.

// Thêm hình ảnh note vào đây

Apache Zeppelin với tích hợp Spark cung cấp:   
     - Tự động SparkContext và SQLContext injection  
     - Tải xuống phụ thuộc vào thời gian chạy của jar từ hệ thống tệp cục bộ hoặc kho lưu trữ maven.  
    - Hủy công việc và hiển thị tiến trình của nó

Apache Zeppelin có thể trực quan hóa dữ liệu dưới nhiều dạng đồ thì khác nhau như : biểu đồ cột, biểu đồ đường, biểu đồ tròn,… với các mục đích khác nhau. Các biểu đồ thì không giới hạn truy vấn SparkSQL. Các note trong Zeppelin có thể xuất ra lưu trữ bên ngoài thuận tiện cho việc sử dụng trên các máy tính khác nhau.

## 2.5 Tableau

Hình ảnh 2.5.1 Logo Tableau Software

Tableau Software là một công ty phần mềm máy tính của Mỹ có trụ sở tại Seattle, WA, Hoa Kỳ. Nó tạo ra các sản phẩm trực quan dữ liệu tương tác tập trung vào BI(business intelligence). Công ty được thành lập tại Khoa Khoa học Máy tính của Đại học Stanford từ năm 1997 đến 2002.

Tableau là một phần mềm hỗ trợ phân tích và trực quan hóa dữ liệu một cách nhanh chóng.

Tableau sử dụng ngôn ngữ truy vấn trực quan VizQL. VizQL là một ngôn ngữ truy vấn trực quan dịch các hoạt động kéo, thả vào các truy vấn dữ liệu và sau đó thể hiện dữ liệu đó một cách trực quan. VizQL mang lại lợi ích đáng kể về khả năng xem và hiểu dữ liệu của mọi người truy vấn và phân tích.

Tableau có nhiều sản phẩm như : Tableau Desktop, Tableau Server, Tableau Online, Table Prep. Trong nghiên cứu này, tôi sử dụng Tableau Desktop. Tableau Desktop là ứng dụng trực quan hóa dữ liệu cho bất kỳ dữ liệu có cấu trúc nào báo cáo, biểu đồ có tương tác cao. Sau khi cài đặt bạn có thể liên kết với các dữ liệu và hiển thị thông tin theo một góc độ đồ họa.

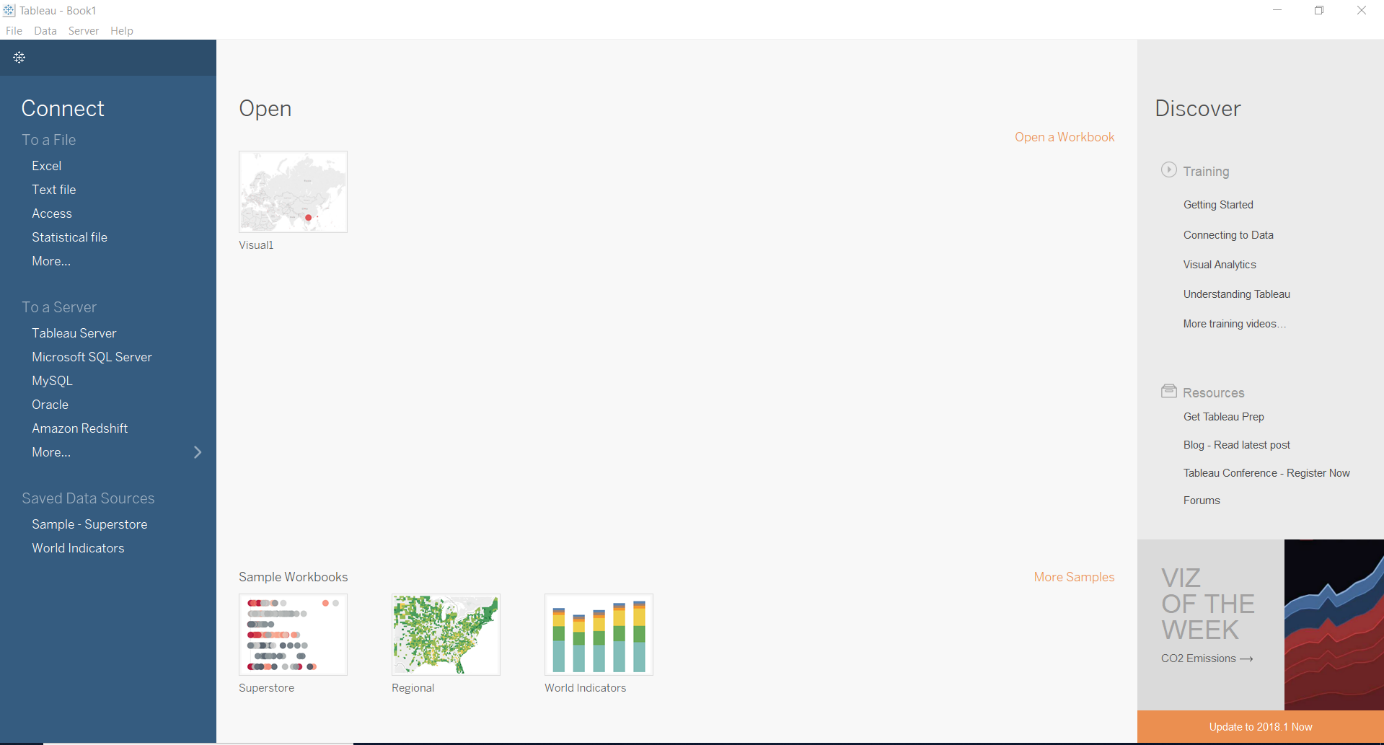
Lợi ích của Tableau:

+ Xử lý kết quả nhanh chóng để có các thông tin hữu ích

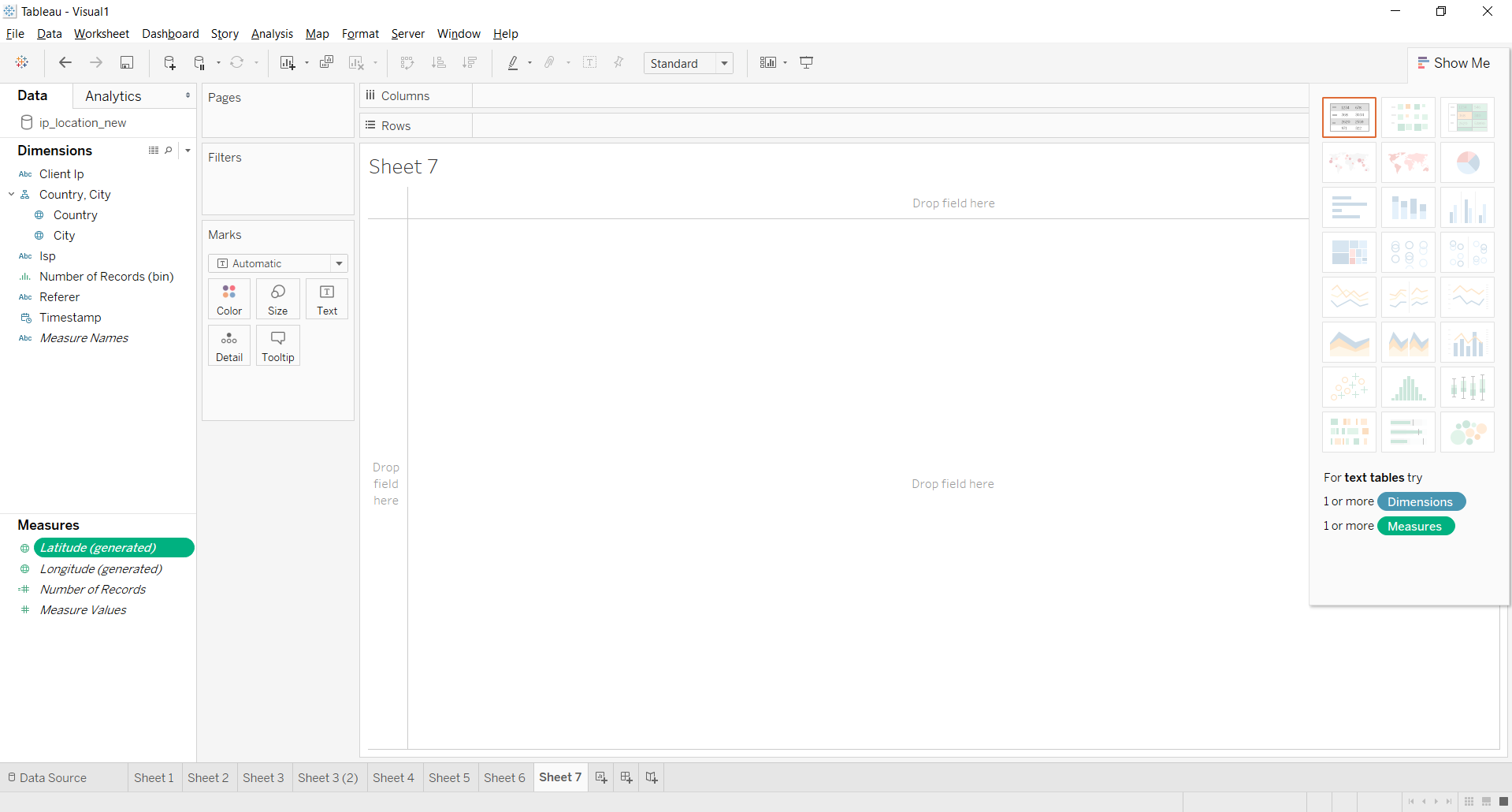
+ Dễ sử dụng cho tất cả mọi người

+ Có thể chèn dữ liệu đầu vào ở nhiều dạng khác nhau:

* Hệ quản trị cơ sở dữ liệu (Oracle, MSSQL)
* Microsoft Access
* Microsoft Excel
* CSV/Flat Files



Hình 2.5.2 Kết nối dữ liệu đầu vào của Tableau Desktop



Hình 2.5.3 Giao diện thực hiện trực quan hóa

Sau khi chèn dữ liệu vào chúng ta có thể thấy hiển thị dự liệu được phân ra các cột. Có các thuộc tính trong dữ liệu thêm vào. Bên phải có “Show Me” cho phép xem các loại biểu đồ sẽ hiển thị. Kéo thả các trường của các cột vào dữ cho tạo ra các biểu đồ theo mục đích sử dụng.

# CHƯƠNG 3 : HIỆN THỰC

## 3.1 Cách thức phân tích log file

Như đã nói ở trên, tôi sẽ phân tích access log của web Server : <http://www.hcmut.edu.vn>.

Log trên ghi lại các hoạt động truy cập của Server của mỗi người truy cập đến. Trong phần nghiên cứu này tôi sẽ công cụ hỗ trợ là Apache Spark và có sử dụng Apache Zeppelin.

Sau khi cài đặt Spark cho Ubuntu thì chúng ta có thể sừ dụng lệnh trên Terminal là spark-shell cho sử dụng ngôn ngữ Scala hoặc pyspark cho ngôn ngữ Python. Bên cạnh đó có thể cài đặt Apache Zeppelin sử dụng trên các trình duyệt để thực hiện kết hợp với Spark.

Log file hiện tại có dung lượng khá lớn với khoảng 21GB. Với dung lượng log file có kích thước khá lớn gây khó khăn cho việc phân tích cùng một lúc mà không có xảy ra lỗi gì.

Log có kích thước lớn và nhìn vào bên trong như một ma trận rộng lớn, gây khó hiểu và không biết phải bắt đầu làm từ đâu để có thể phân tích ra. Sau đó, tôi thực hiện chia nhỏ file ra để có thể nghiên cứu chi tiết về từng dòng được ghi trong file. Qua đó có thể biết được cấu trúc bên trong như thế nào. Tìm hiểu rõ hơn về từng phần trong cấu trúc đó. Từng dòng trong file log được chia ra các phần nhỏ và chi tiết hơn được nêu ở phần 2.2. Qua các phần chi tiết như thế, tôi sẽ tạo ra một lớp chứa các trường trong log. Để có thể chuyển các thành phần trong log như thế thì cần phải chú ý hơn về giữa các thành phần, chúng được phân ra với nhau bởi một dấu cách. Như thế chúng ta chia ra theo dấu cách thì sẽ được các phần nhỏ hơn để đưa vào các trường có trong lớp mới đã được tạo ra. Chúng ta có thể ra một dữ liệu mới được định dạng thành các trường cụ thể hơn dễ cho việc truy xuất dữ liệu phục vụ cho các phần tiếp theo.

Bên trong các dòng log có khá nhiều lỗi như: dung lượng chuyển file là dấu -; các dòng tham khảo đến từ nguồn nào thì thay vào đó là chuỗi kí tự không rõ hay cũng có thể là đường dẫn đến một file định d0ạng trang web,..…

// thêm hình về dòng log bị lỗi

Để có thể lọc bỏ các phần bị lỗi, tôi dùng đến các điều kiện so sánh ở các phần đó trước khi được thực hiện tiếp. Ngoài ra còn sử dụng các biểu thức chính quy để loại bỏ nhanh hơn về các đường dẫn là các file định dạng web. Khi phân tích từ log file thô ban đầu, tôi lọc bỏ đi các phần bị lỗi làm hữu ích hơn dữ liệu được tạo ra tránh gây nhiễu cho các bước làm tiếp. Trong phần thực hiện này tôi chú ý đến các phần chính như : địa chỉ IP, đường dẫn tham khảo, thời gian của từng dòng log,..

Qua bước lọai bỏ thứ nhất, chúng ta sẽ giảm bớt đi các dòng lỗi bên trong đó. Thực hiện tiếp theo muốn tạo ra các tập dữ liệu có ích khác.

Việc phân tích ban đầu để hiểu rõ hơn về các nội dung như thế nào trong log file. Tôi muốn tạo ra tập chứ các đường dẫn tham khảo nào được truy cập như thế nào. Mỗi người dùng có thể truy cập vào web đó nhiều lần khác nhau với nhau, đơn nhiên là có cùng nội dung. Tôi sẽ thống kê lại các đường dẫn nào được truy cập bao nhiêu. Từ dữ liệu mới có chúng ta làm tiếp thực hiện quét qua các dòng trong dữ liệu nếu phần đường dẫn nào giống nhau thì sẽ tăng lên.

Nhưng khi quét qua các dòng như thế đối với các dữ liệu nhỏ khoảng vài trăm, vài ngàn dòng thì không sao hết, nhưng log này có kích thước lớn với khoảng lên cả chục triệu dòng sẽ gây khó khăn hơn về thời gian xử lý.Qua phần tìm hiểu Spark sử dụng ngôn ngữ Scala có nhiều hàm hỗ trợ cho chúng ta phân tích nhanh hơn để đạt được mục đích là có tập về số lượng truy cập của các đường dẫn tham khảo.Tôi sử dụng hàm map được tích hợp sẵn trong Scala và thực thi theo cơ chế song song phân chia lại các phần trong tập dữ liệu mới có đó. Tôi chỉ lấy phần đường dẫn tham khảo ở các dòng trong log để đem đi so sánh vào đếm có số lần truy cập. Nếu mỗi dòng mới là lại phải duyệt xem nó đã có trước đó chưa là tăng lên hay là thêm nó vào nội dung mới. Thấy như vậy có tìm hiểu thêm để thực hiện sao nhanh hơn. Sau

đó tôi tìm thấy hàm reduceByKey, hàm đó giúp chúng ta duyệt lại các nội dung nhanh chóng và tăng lên số lần nếu nội dung đó đã được nêu trước đó. Chúng ta có thể sắp xếp lại tập dữ liệu để hiển thị xem đường dẫn nào được tham khảo nhiều nhất và giảm dần theo sau đó. Định danh có các đường tham khao khảo để có thể hiển thị gọn hơn cho các đường dẫn. Chúng ta đã sắp theo số lần truy cập giảm dần nên sẽ biết rõ định danh đầu là dường dẫn tham khảo nhiều nhất và giảm dần theo đó còn định danh sẽ tăng dần theo thứ tự. Như vậy ta có được tập dự liệu về thống kê số lần truy cập từ các đường tham khảo.

//https://spark.apache.org/docs/2.2.1/rdd-programming-guide.html

Tạo ra output thống kê số lần truy cập của các nội dung trong website. Tôi sẽ thực hiện quét các dòng trong log. So sánh có nội dung nào giống nhau thì sẽ tăng lên biến đếm.

Lọc ra các cột có ích hơn log để tạo ra dữ liệu mới nhỏ hơn tiện cho việc xử lý. Như tạo ra output có cột IP address, Timestamp, Referrer URL.

Từ dữ liệu mới đó có thể tạo ra các output mới như: Các nội dung truy cập trong 1 ngày, 1 giờ; trong 1 ngày Ip nào truy cập bao nhiêu lần;…

Từ các IP có thể biết được vị trí của nơi nào truy cập đến website. Tìm ra các quốc gia, thành phố và ISP của các IP đó.

## 3.2 Các trực quan hóa dữ liệu