VIETNAM NATIONAL UNIVERSITY HCM

UNIVERSITY OF INFORMATION TECHNOLOGY

FACULTY OF COMPUTER NETWORKS AND COMMUNICATIONS

BÁO CÁO ĐỒ ÁN

NGHIÊN CỨU, TRIỂN KHAI DISTRIBUTED SYSTEM APACHE SPARK

GVHD: ThS. ĐẶNG LÊ BẢO CHƯƠNG

Lớp: NT531.N21

Nhóm 4

Nguyễn Duy Trọng Nhân – 20520669

Mai Phước Sang – 20520735

Thành phố Hồ Chí Minh, ngày 10 tháng 06 năm 2023

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

……, ngày…….....tháng……năm 2023

Người nhận xét

(Ký tên và ghi rõ họ tên)

NỘI DUNG

A. DANH MỤC CÁC BẢNG, HÌNH VÀ BIỂU ĐỒ

B. CƠ SỞ LÝ THUYẾT

I) Khái niệm và tổng quát

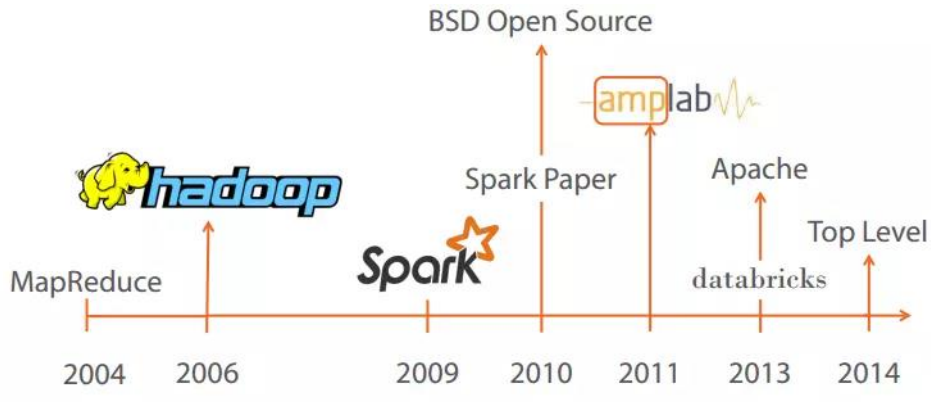
1) Tính cần thiết cho sự ra đời của hệ thống, công cụ xử lý dữ liệu cho BigData

- Sự bùng nổ của lượng dữ liệu khổng lồ cùng sự xuất hiện của các trung tâm dữ liệu người dùng phía các dịch vụ Youtube, Facebook, Google,…
- Nhu cầu xử lý khối lượng dữ liệu lớn, đa dạng và phức tạp, trong thời gian thực.
- Nhu cầu thu thập và phân tích dữ liệu để đưa đến các quyết định và động thái chiến lược kinh doanh.

2) Apache Spark

- Apache Spark là một hệ thống xử lý phân tán mã nguồn mở được sử dụng cho việc xử lý Big-Data, được phát triển vào năm 2009 bởi AMPLab,tiếp đó đã được trao cho Apache Software Foundation vào năm 2013 và tiếp tục được phát triển cho đến nay.[1]

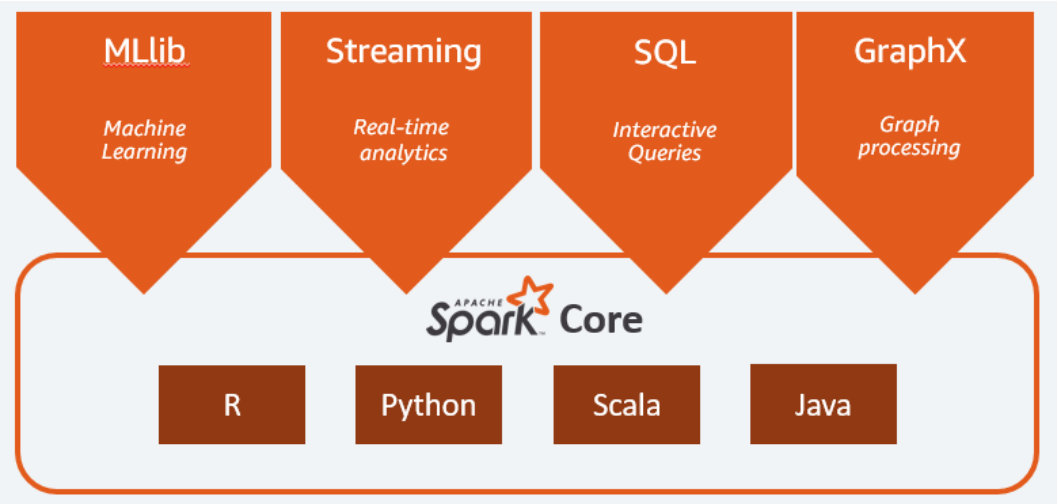
- Nó sử dụng bộ nhớ cache để tối ưu tốc độ thực hiện các truy vấn và phân tích dữ liệu với bất kỳ kích thước nào.
- Nó cung cấp các API phát triển bằng Java, Scala, Python và R, và hỗ trợ việc tái sử dụng mã nguồn cho nhiều công việc khác nhau như xử lý lô, truy vấn tương tác, phân tích thời gian thực, học máy và xử lý đồ thị.
- Được sử dụng bởi nhiều tổ chức lớn thuộc đa ngành nghề, Apache Spark đã trở thành một trong những framework xử lý dữ liệu phân tán dữ liệu lớn phổ biến nhất với 365.000 thành viên meetup vào năm 2017.[2]



Hình 1: Sự phát triển của các công cụ xử lý dữ liệu

II) Tổng quan về Apache Spark

1) Các thành phần của Apache Spark



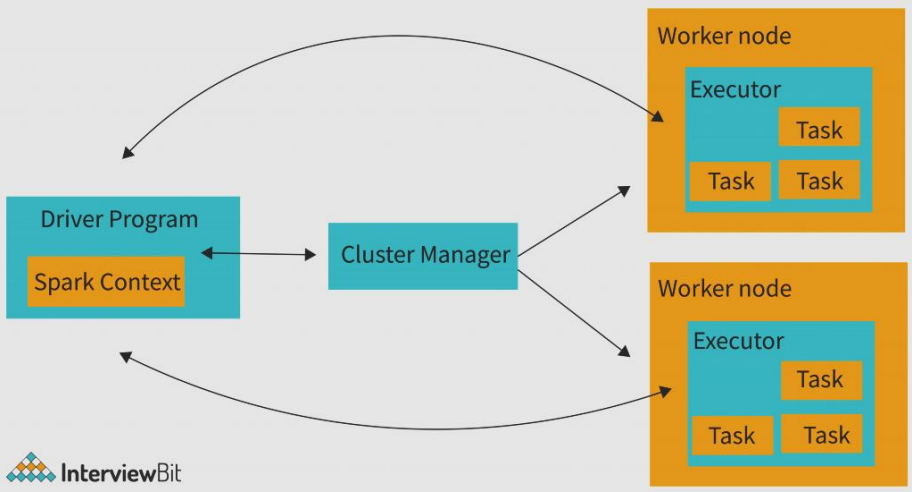
Hình 2: Thành phần của Apache Spark

- Apache Spark có 5 phần chính:
    - Spark Core
    - SparkSQL
    - Spark Streaming
    - MLib
    - GraphX
- Spark Core: Đây được xem là nền tảng và điều kiện cho sự vân hành của hệ thông Apache Spark. Nó chịu trách nhiệm quản lý bộ nhớ, khôi phục lỗi, lập lịch, phân phối và giám sát công việc và tương tác với các hệ thống lưu trữ. Đặc biệt, Spark Core cung cấp API để định nghĩa RDD (Resilient Distributed DataSet) - tập hợp của các item được phân tán trên các node của cluster và có thể được xử lý song song.[3]
- RDDs là một loại dữ liệu cơ bản trong Apache Spark và được sử dụng để xử lý các task phân tán. RDDs có thể được tạo ra bằng cách đọc dữ liệu từ các nguồn bên ngoài, hoặc bằng cách chuyển đổi và tiền xử lý dữ liệu có sẵn trong các RDD khác. Tính năng quan trọng của RDD là tự phục hồi, vì vậy nếu một node trong cluster bị sập, các phần tử trong RDD vẫn có thể được tái tạo trên các node khác của Spark cluster.[4]
- SparkSQL: Spark SQL là một công cụ truy vấn phân tán có các đặc điểm như:truy vấn tương tác, độ trễ thấp, tốc độ xử lý nhanh hơn 100 lần so với MapReduce. SparkSQL cho phép truy vấn dữ liệu cấu trúc và nửa cấu trúc (semi-structured data), hỗ trợ thao tác với nhiều nguồn dữ liệu như: JDBC, ODBC, JSON, HDFS, Hive, ORC, và Parquet.
- Spark Streaming: là giải pháp xử lý real-time tận dụng khả năng lập lịch nhanh của SparkCore để phân tích streaming. Ở đầu vào, dữ liệu được chia nhỏ thành từng mini-batch và có thể được xử lý, phân tích bằng source code của việc xử lý batch.
- MLib: là một nền tảng học máy phân tán dựa trên dữ liệu quy mô lớn. Các mô hình Machine Learning có thể được đào tạo bằng các ngôn ngữ R hoặc

Python. MLib cung cấp nhiều thuật toán của học máy như: classification, regression, clustering, collaborative filtering…

- GraphX: Spark GraphX là một framework xử lý đồ thị phân tán cho phép người dùng tương tác xây dựng và chuyển đổi cấu trúc dữ liệu đồ thị trên quy mô lớn.

2) Kiến trúc của Apache Spark



Hình 3: Kiến trúc của Apache Spark

- Trong Apache Spark, kiến trúc để xử lý dữ liệu gồm 3 phần quan trọng: Cluster Manager, Spark Context và Worker Node.[3]
- Spark có thể chạy trên nhiều loại Cluster Managers như Hadoop YARN, Apache Mesos, Kubernetes hoặc trên chính cluster manager được tích hợp Spark được gọi là Standalone Scheduler.
- Spark Context được tạo ra để cung cấp quyền truy cập vào tất cả các chức năng của Spark. Spark sẽ thực thi những gì có trong source code thông qua Spark context.

- Một job trong Spark sẽ được chia thành nhiều task và phân phối đến các worker node khác nhau.
- Node chứa Spark context được gọi là master node. Spark context và cluster manager phối hợp xử lý các công việc phức tạp khác nhau. Các worker node trong Spark thực thi các nhiệm vụ đa luồng trên các RDD đã được phân vùng và kết quả sẽ được trả về cho Spark Context.
- Trong suốt quá trình thực thi job, Spark Context luôn giám sát và nhận các kết nối đến từ các worker node.

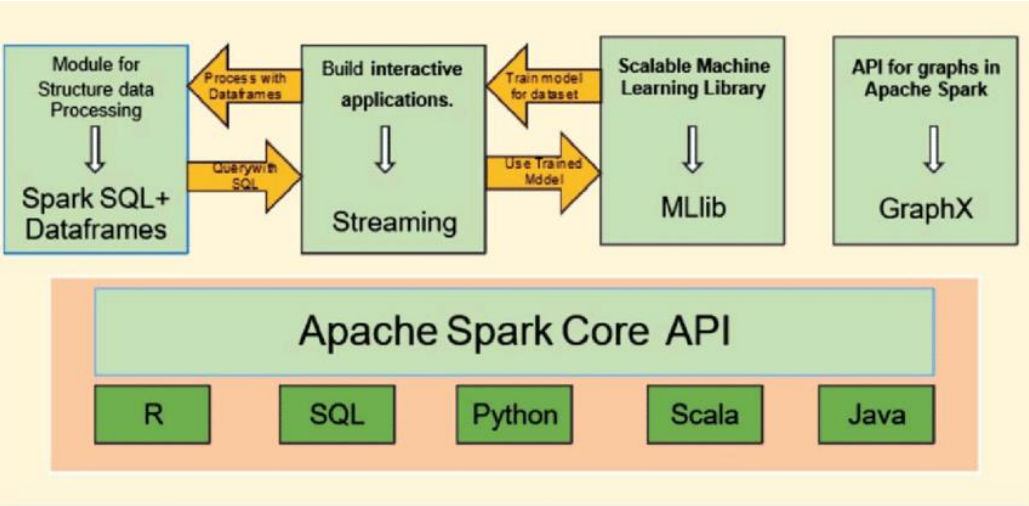
3) Apache Spark hoạt động như thế nào

- Spark được tạo ra để giải quyết các hạn chế đối với MapReduce, bằng cách xử lý trong bộ nhớ, giảm số bước trong một tác vụ và tái sử dụng dữ liệu trên nhiều hoạt động song song.
- Với Spark, chỉ cần một bước: dữ liệu được đọc vào bộ nhớ - xử lý dữ liệu và ghi lại kết quả — điều này dẫn đến việc thực thi nhanh hơn nhiều. Spark cũng tái sử dụng dữ liệu bằng cách sử dụng bộ nhớ đệm trong bộ nhớ để tăng tốc đáng kể các thuật toán ML mà liên tục gọi một hàm trên cùng một dataset.
- Việc tái sử dụng dữ liệu được thực hiện thông qua việc tạo DataFrames, một khái niêm trên Tập dữ liệu phân tán có khả năng phục hồi (RDD), là một tập hợp các đối tượng được lưu trong bộ nhớ và được sử dụng lại trong nhiều lần. Điều này làm giảm đáng kể độ trễ làm cho Spark nhanh hơn nhiều lần so với MapReduce, đặc biệt là khi thực hiện học máy và phân tích tương tác.[1]

4) Những điểm nổi bật của Apache Spark

- “Spark as a Service”: Giao diện REST để quản lí (submit, start, stop, xem trạng thái) spark job, spark context
- Tăng tốc, giảm độ trễ thực thi job xuống mức chỉ tính bằng giây bằng cách tạo sẵn spark context cho các job dùng chung.
- Dễ dàng tích hợp với các công cụ báo cáo như: Business Intelligence, Analytics, Data Integration Tools.

- Fault tolerance: Spark đạt được khả năng này nhờ vào sử dụng DAG và RDD (Resilient Distributed Datasets). DAG ở đây chứa tất cả các bước (step) cần thiết để hoàn thành một task. Tất cả đều được ghi lại. Chính vì vậy, khi xảy ra lỗi ở nodes worker nào đó, ta có thể tái hiện lỗi từ DAG đã lưu hiện có.[5]
- Apache Spark tích hợp đa ngôn ngữ. Hầu hết đều có API cho các ngôn ngữ Java, Scala, Python và R. Ngoài ra, còn có các tính năng nâng cao với ngôn ngữ R để phân tích dữ liệu. Ngoài ra Spark có bộ SQL riêng là Spark SQL, tương tự với SQL.



Hình 4: Sự tương tác giữa các thành phần của Apache Spark

5) So sánh Apache Spark và Apache Hadoop

- Hiệu năng:
  - Spark được cho là nhanh hơn Hadoop gấp 100 lần khi chạy trên RAM, và gấp 10 lần khi chạy trên ổ cứng.
  - Người ta cho rằng Spark sắp xếp 100TB dữ liệu nhanh gấp 3 lần Hadoop trong khi sử dụng ít hơn 10 lần số lượng hệ thống máy tính.
  - Nhờ xử lý ở bộ nhớ RAM nên Spark cung cấp các phân tích dữ liệu real-time cho các chiến dịch quảng cáo, ML (học máy), hay các trang web mạng xã hội.
  - Tuy nhiên, khi Spark làm việc cùng các dịch vụ chia sẻ khác chạy trên YARN thì hiệu năng có thể giảm xuống. Hadoop thì khác, nó dễ

dàng xử lý vấn đề này. Nếu người dùng có khuynh hướng xử lý hàng loạt (batch process) thì Hadoop lại hiệu quả hơn Spark.

- Bảo mật:
    - Hiện tại, Spark chỉ hỗ trợ xác thực mật khẩu (password authentication).
    - Hadoop trang bị toàn bộ các mức độ bảo mật như Hadoop Authentication, Hadoop Authorization, Hadoop Auditing, và Hadoop Encryption.
- Chi phí:
    - Đều là các framework mã nguồn mở (open source).
    - Spark cần một lượng lớn RAM vì nó xử lý mọi thứ ở bộ nhớ. Do đó, đây là yếu tố ảnh hưởng đến chi phí vì giá RAM cao hơn ổ đĩa.
    - Hadoop chỉ sử dụng ổ đĩa nhưng lại cần nhiều hệ thống để phân bổ tài nguyên ổ đĩa[6]

Vì vậy, việc lựa chọn framework nào phụ thuộc yêu cầu cụ thể từng dự án. Nếu nhu cầu là xử lý lượng lớn dữ liệu dạng lịch sử thì Hadoop là lựa chọn tốt vì dữ liệu dạng này cần lưu và có thể được xử lý trên ổ đĩa. Còn khi yêu cầu là xử lý dữ liệu thời gian thực thì Spark là lựa chọn tối ưu vì ta chỉ cần ít hệ thống cho sử lý cùng một lượng lớn dữ liệu với thời gian giảm nhiều lần hơn khi sử dụng Hadoop.

6) Các use case của Apache Spark

- Spark đã được triển khai trong mọi loại tình huống sử dụng dữ liệu lớn để phát hiện mẫu và cung cấp thông tin thời gian thực. Các ví dụ về các use case như:
    - Dịch vụ tài chính: Spark được sử dụng trong ngành ngân hàng để dự đoán việc tiếp tục sử dụng dịch vụ của khách hàng và đề xuất các sản phẩm tài chính mới. Trong ngành ngân hàng đầu tư, Spark được sử dụng để phân tích giá cổ phiếu để dự đoán xu hướng tương lai.
    - Chăm sóc sức khỏe: Spark được sử dụng để xây dựng chăm sóc toàn diện cho bệnh nhân bằng cách cung cấp dữ liệu cho các nhân viên y

tế trực tiếp trong mọi tương tác với bệnh nhân. Spark cũng có thể được sử dụng để dự đoán và gợi ý liệu trình điều trị cho bệnh nhân.

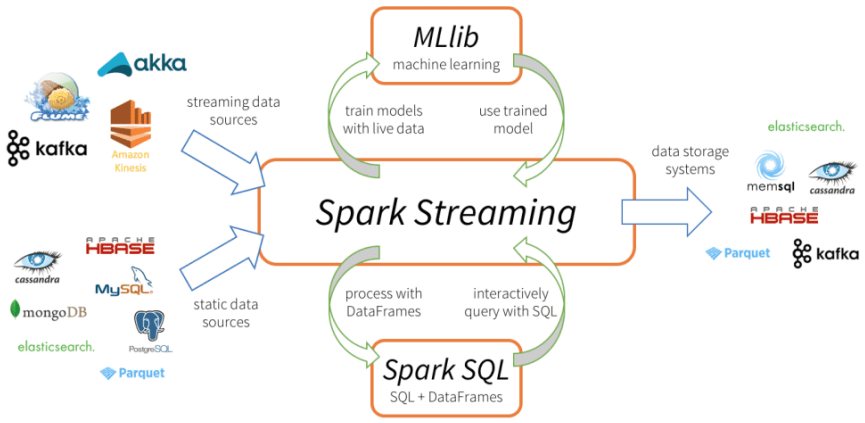
- Sản xuất: Spark được sử dụng để loại bỏ thời gian chết của các thiết bị kết nối internet bằng cách đề xuất thời điểm bảo trì định kỳ.
- Bán lẻ: Spark được sử dụng để thu hút và giữ chân khách hàng thông qua các dịch vụ và ưu đãi cá nhân.

III) Apache Spark Streaming

1) Dữ liệu streaming là gì

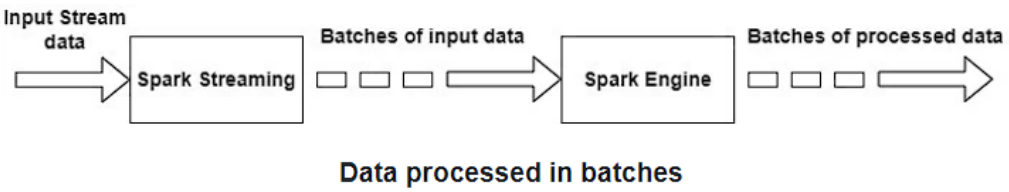
- Dữ liệu streaming là 1 loại dữ liệu liên tục được tạo từ các nguồn như các sensor IOT, từ các web Server hay từ các tìm kiếm trên internet. Dữ liệu này được truyền và xử lý trong thời gian thực, giúp chúng ta theo dõi và phân tích các sự kiện đang diễn ra ngay lập tức.
- Dữ liệu streaming thường có tính chất động và nhanh chóng dồi dào, tức là lượng dữ liệu của nó thay đổi liên tục và rất lớn. Vì những tính chất của loại dữ liệu này, ngành công nghiệp dữ liệu streaming hiện nay có giá trị lên đến hàng trăm tỷ đô-la. [7]
- Nếu được xử lý và phân tích đúng cách, các dữ liệu streaming có thể đem lại cho công ty nhiều lợi ích, từ việc theo dõi sự kiện trong thời gian thực, điều hành các kịch bản tự động hóa cho đến cung cấp cho khách hàng trải nghiệm tốt hơn và các sản phẩm dịch vụ tốt hơn.

2) Spark Streaming xử lý dữ liệu streaming như thế nào



Hình 5: Mô hình của Apache Spark Streaming

- Spark Streaming sẽ xử lý dữ liệu theo 3 giai đoạn chính:
    - Nguồn dữ liệu: đến từ các nguồn dữ liệu streaming như Kafka,Flume, Kinesis,.. , các nguồn dữ liệu tĩnh như MySQL, MongoDB, Cassandra,…, TCP Sockets,…
    - Chia các luồng dữ liệu streaming thành các batch (còn gọi là micro-batch và sau đó xử lý mỗi lô dữ liệu dưới dạng RDD.
    - Spark Streaming engine: sau đó xử lý các RDD bằng các hàm tích hợp, thuật toán phức tạp như map, reduce, reduceByKey, join và window. Ngoài ra, chúng ta có thể truy vấn các luồng trực tiếp, áp dụng machine learning SparkSQL và MLlib.[7]
    - Output: Kết quả sẽ được trả về theo batch, sau đó được chuyển đến các database, file system hoặc trực quan hóa bằng dashboard,…



Hình 6: Luồng xử lý của Apache Spark Streaming

3) Các thành phần

Sau đây là các thành phần tạo nên 1 ứng dụng Spark Streaming

- DStream: hay đầy đủ là Discretized Stream , là 1 khái niệm trong Spark Streaming. Nó là 1 chuỗi RDD có tính liên tục, được tạo ra từ các nguồn dữ liệu streaming hay từ chính các phép toán được thực hiện từ các DStream khác như: map, flatMap, count, countByValue, updateStateByKey,..
- StreamingContext: kế thừa và phụ thuộc vào SparkContext, StreamingContext là 1 entrypoint cho tất cả chức năng xử lý dữ liệu real-time. Nó được sử dụng để thiết lập kết nối đến các nguồn dữ liệu đầu vào và thiết lập cả các phép xử lý, biến đổi trên luồng dữ liệu.

- Spark Streaming API: cung cấp vài API của các ngôn ngữ được hỗ trợ Java API, Python API và Spark Streaming Scala API. Nó cung cấp khả năng xử lý dữ liệu streaming và fault-tolerant. Nó cho phép xử lý dữ liệu đầu vào từ nhiều nguồn như Kafka, Kinesis hoặc TCP sockets và có thể được xử lý bằng các thuật toán phức tạp như map, reduce, join và window.

4) Cách 1 ứng dụng Spark Streaming hoạt động

- Spark Streaming Context được sử dụng để xử lý dữ liệu luồng thời gian thực. Vì vậy, bước đầu tiên là khởi tạo đối tượng StreamingContext bằng hai tham số: SparkContext và chu kỳ cho batch. Sau khi StreamingContext được khởi tạo, không thể định nghĩa hoặc thêm các tính toán mới vào Context

hiện có. Ngoài ra, chỉ có một đối tượng StreamingContext có thể hoạt động cùng một lúc.

- Sau khi định nghĩa Spark StreamingContext, chúng ta xác định các nguồn dữ liệu đầu vào bằng cách tạo các đối tượng DStreams.
- Xác định các tính toán sử dụng Spark Streaming Transformations API như map và reduce cho DStreams.
- Sau khi định nghĩa các logic tính toán, chúng ta có thể bắt đầu nhận dữ liệu và xử lý nó bằng cách sử dụng phương thức start trong đối tượng StreamingContext đã được tạo trước đó.
- Cuối cùng, chúng ta đợi việc xử lý dữ liệu luồng được dừng bằng cách sử dụng phương thức awaitTermination của đối tượng StreamingContext.

5) Use case

- Spark Streaming đang trở thành nền tảng được ưa chuộng để triển khai các giải pháp xử lý dữ liệu và phân tích cho dữ liệu thời gian thực nhận được từ Internet of Things (IoT) và cảm biến. Nó được sử dụng trong nhiều ngành công nghiệp và trường hợp sử dụng đáng chú ý như:
    - Phân tích chuỗi cung ứng: Spark Streaming có thể được sử dụng để phân tích dữ liệu thời gian thực từ chuỗi cung ứng, giúp doanh nghiệp theo dõi tồn kho, theo dõi lô hàng, tối ưu hóa vận chuyển và đưa ra quyết định dựa trên dữ liệu để cải thiện hiệu quả và sự hài lòng của khách hàng.
    - Phân tích video thời gian thực: Các nền tảng video trực tuyến có thể sử dụng Spark Streaming để phân tích tương tác của người xem, giám sát hiệu suất video và cung cấp trải nghiệm cá nhân và tương tác thời gian thực. Nó cho phép gợi ý thời gian thực, cá nhân hóa nội dung và quảng cáo đích theo hành vi người dùng.
- Và sau đây là các ví dụ cho Spark Streaming được các công ty công nghệ ứng dụng và triển khai:
    - Uber: Uber sử dụng Spark Streaming để thu thập hàng terabyte dữ liệu sự kiện mỗi ngày từ người dùng để phân tích dữ liệu thời gian

thực. Điều này giúp họ hiểu hành vi người dùng, tối ưu hóa dịch vụ và nâng cao trải nghiệm người dùng chung.

- Pinterest: tận dụng Spark Streaming, cùng với MemSQL và công nghệ Apache Kafka. Họ sử dụng sự kết hợp này để có cái nhìn thời gian thực về cách người dùng của họ tương tác trên toàn cầu. Điều này giúp họ hiểu được sở thích của người dùng, cải thiện gợi ý nội dung và nâng cao sự tương tác của người dùng trên nền tảng của họ.
- Netflix: Netflix sử dụng Kafka và Spark Streaming để xây dựng một giải pháp gợi ý phim trực tuyến và giám sát dữ liệu thời gian thực. Họ xử lý hàng tỷ sự kiện hàng ngày từ các nguồn dữ liệu khác nhau để cung cấp gợi ý phim cá nhân cho người dùng. Ngoài ra, Spark Streaming cũng hỗ trợ giám sát và phân tích dữ liệu theo thời gian thực, giúp tối ưu hoạt động và duy trì chất lượng dịch vụ.

C. THAM KHẢO, TRÍCH DẪN

[1] "What is Apache Spark? | Introduction to Apache Spark and Analytics | AWS," *Amazon Web Services, Inc.* https://aws.amazon.com/big-data/what-is-spark/.

[2] "Spark Streaming - Spark 3.4.1 Documentation." https://spark.apache.org/docs/latest/streaming-programming-guide.html .

[3] "Apache Spark Architecture - Detailed Explanation," *InterviewBit*, Jun. 03, 2022. https://www.interviewbit.com/blog/apache-spark-architecture/.

[4] A. Team, "Apache Spark Architecture Detail Explained.," *AnalyticsLearn*, Dec. 06, 2020. https://analyticslearn.com/apache-spark-architecture-detail-explained.

[5] B. biên tập TopDev, "10 tính năng trên Apache Spark anh em nên biết," *TopDev*, May 05, 2023. https://topdev.vn/blog/10-tinh-nang-tren-apache-spark/.

[6] "Hadoop và Spark Big data framework nào tốt nhất cho bạn," Jul. 22, 2019. https://viblo.asia/p/hadoop-va-spark-big-data-framework-nao-tot-nhat-cho-ban-4dbZNqRqKYM.

[7] "Big Data Processing with Apache Spark - Part 3: Spark Streaming." https://www.infoq.com/articles/apache-spark-streaming/.

Link demo: https://drive.google.com/drive/folders/1NvynAzM1LQ2xNAkq2GOeMJ8SX8jtsXjA?usp=drive_link