TON DUC THANG UNIVERSITY

UNI]]

MINING OF

MASSIVE

DATASETS

NGUYEN HAI LONG

**MỤC LỤC**

[Chương I: Tìm hiểu về Dataframe 1](#_Toc70881219)

[**1.** **Giới thiệu** 1](#_Toc70881220)

[**2.** **Tại sao chúng ta cần sử dụng Dataframe** 1](#_Toc70881221)

[***2.1*** ***Xử lý dữ liệu có cấu trúc và bán cấu trúc*** 1](#_Toc70881222)

[***2.2*** ***Cắt và thái lát dữ liệu*** 1](#_Toc70881223)

[***2.3*** ***Nguồn dữ liệu*** 1](#_Toc70881224)

[***2.4*** ***Hỗ trợ nhiều ngôn ngữ*** 1](#_Toc70881225)

[**3.** **Các tính năng của Dataframe** 2](#_Toc70881226)

[***3.1*** ***Phân phối*** 2](#_Toc70881227)

[***3.2*** ***Đánh giá lười biếng*** 2](#_Toc70881228)

[***3.3*** ***Bất biến*** 2](#_Toc70881229)

[***3.4*** ***Tập hợp phân tán của đối tượng Row*** 2](#_Toc70881230)

[***3.5*** ***Xử lý dữ liệu*** 2](#_Toc70881231)

[***3.6*** ***Tối ưu hóa bằng cách sử dụng trình tối ưu hóa chất xúc tác*** 2](#_Toc70881232)

[***3.7*** ***Khả năng tương tác Hive*** 2](#_Toc70881233)

[***3.8*** ***Vonfram*** 2](#_Toc70881234)

[**4.** **Hạn chế của Dataframe** 3](#_Toc70881235)

[***4.1*** ***An toàn thời gian biên dịch*** 3](#_Toc70881236)

[***4.2*** ***Không thể hoạt động trên đối tượng miền (đối tượng miền bị mất)*** 3](#_Toc70881237)

[Tài liệu tham khảo 4](#_Toc70881238)

# Chương I: Tìm hiểu về Dataframe

1. **Giới thiệu**

DataFrames là một từ thông dụng trong ngành công nghiệp ngày nay thường đề cập đến một cấu trúc dữ liệu, về bản chất là dạng bảng. Nó đại diện cho các hàng, mỗi hàng bao gồm một số quan sát. Các hàng có thể có nhiều định dạng dữ liệu (không đồng nhất), trong khi một cột có thể có dữ liệu cùng loại (đồng nhất). DataFrames thường chứa một số siêu dữ liệu ngoài dữ liệu như tên cột và hàng.

Spark Dataframe có thể thiết lập Schema cho Spark RDD và có thể tạo Object Dataframe. Giống như viết SQL, Spark Dataframe đầy đủ chức năng như select, where ... đặc biệt là join với các DataFrame khác; sử dụng các method như filter, select để trích xuất dữ liệu theo cột, hàng; xử gọn các loại data như Log ... với groupBy → agg; thêm 1 cột dễ dàng với UDF(User Defined Function). Ngoài ra Spark DataFrame cũng đã hỗ trợ Pivot (Spark 1.6 trở lên) rất hữu ích cho việc lập bảng biểu, báo cáo.[1]

Spark Dataframe rất dễ xài, đơn giản mà hiệu suất và khả năng truy vấn tốt hơn RDD. Trong các trường hợp thông thường thì bạn nên xài Dataframe.

1. **Tại sao chúng ta cần sử dụng Dataframe**
   1. ***Xử lý dữ liệu có cấu trúc và bán cấu trúc***

DataFrames được thiết kế để xử lý một tập hợp lớn dữ liệu có cấu trúc cũng như bán cấu trúc. Các quan sát trong Spark DataFrame được tổ chức dưới các cột được đặt tên, giúp Apache Spark hiểu sơ đồ của Dataframe. Điều này giúp Spark tối ưu hóa kế hoạch thực hiện trên các truy vấn này. Nó cũng có thể xử lý petabyte dữ liệu.[3]

* 1. ***Cắt và thái lát dữ liệu***

API DataFrames thường hỗ trợ các phương thức phức tạp để cắt và xử lý dữ liệu. Nó bao gồm các hoạt động như "chọn" các hàng, cột và ô theo tên hoặc theo số, lọc ra các hàng, v.v. Dữ liệu thống kê thường rất lộn xộn và chứa nhiều giá trị thiếu và không chính xác và vi phạm phạm vi. Vì vậy, một tính năng cực kỳ quan trọng của DataFrames là quản lý rõ ràng dữ liệu bị thiếu.[3]

* 1. ***Nguồn dữ liệu***

DataFrames đã hỗ trợ cho một loạt các định dạng và nguồn dữ liệu. Họ có thể lấy dữ liệu từ nhiều nguồn khác nhau.[3]

* 1. ***Hỗ trợ nhiều ngôn ngữ***

Dataframe có hỗ trợ API cho các ngôn ngữ khác nhau như Python, R, Scala, Java, giúp mọi người có nền tảng lập trình khác nhau dễ sử dụng hơn.[3]

1. **Các tính năng của Dataframe**
   1. ***Phân phối***

DataFrames được phân phối trong tự nhiên, làm cho nó có cấu trúc dữ liệu có khả năng chịu lỗi và có tính sẵn sàng cao.

* 1. ***Đánh giá lười biếng***

Đánh giá lười biếng là một chiến lược đánh giá giữ đánh giá biểu thức cho đến khi cần giá trị của nó. Nó tránh đánh giá lặp đi lặp lại. Đánh giá lười biếng trong Spark có nghĩa là việc thực thi sẽ không bắt đầu cho đến khi một hành động được kích hoạt. Trong Spark, hình ảnh đánh giá lười biếng xuất hiện khi biến đổi Spark xảy ra.

* 1. ***Bất biến***

DataFrames là bất biến trong tự nhiên. Bằng cách bất biến, nghĩa là nó là một đối tượng có trạng thái không thể sửa đổi sau khi nó được tạo. Nhưng chúng ta có thể biến đổi các giá trị của nó bằng cách áp dụng một phép biến đổi nhất định, như trong RDD.

* 1. ***Tập hợp phân tán của đối tượng Row***

DataFrame là tập hợp dữ liệu phân tán được sắp xếp thành các cột được đặt tên. Nó là khái niệm tương đương với một bảng trong cơ sở dữ liệu quan hệ, nhưng tối ưu hóa và phong phú hơn.[2]

* 1. ***Xử lý dữ liệu***

Xử lý các định dạng dữ liệu có cấu trúc và không cấu trúc (Avro, CSV, tìm kiếm đàn hồi và Cassandra) và các hệ thống lưu trữ (HDFS, bảng HIVE, MySQL, v.v.). Nó có thể đọc và viết từ tất cả các nguồn dữ liệu khác nhau.[2]

* 1. ***Tối ưu hóa bằng cách sử dụng trình tối ưu hóa chất xúc tác***

Nó cung cấp năng lượng cho cả truy vấn SQL và API DataFrame. Dataframe sử dụng khung chuyển đổi cây xúc tác theo bốn giai đoạn:[2]

* Phân tích một kế hoạch hợp lý để giải quyết các tham chiếu.
* Tối ưu hóa kế hoạch logic.
* Lập kế hoạch vật lý.
* Tạo mã để biên dịch các phần của truy vấn sang Java bytecode.
  1. ***Khả năng tương tác Hive***

Sử dụng Spark SQL, bạn có thể chạy các truy vấn Hive chưa sửa đổi trên kho Hive hiện tại của bạn. Nó sử dụng lại lối vào Hive và MetaStore và cung cấp cho bạn khả năng tương thích hoàn toàn với dữ liệu, truy vấn và UDF hiện có.[2]

* 1. ***Vonfram***

Vonfram cung cấp một phụ trợ thực thi vật lý, trong đó quản lý bộ nhớ một cách tự nhiên và tự động tạo mã byte để đánh giá biểu thức.[2]

1. **Hạn chế của Dataframe**
   1. ***An toàn thời gian biên dịch***

API Dataframe không hỗ trợ an toàn thời gian biên dịch, điều này hạn chế bạn thao tác dữ liệu không cấu trúc. Tuy nhiên, bạn sẽ nhận được một ngoại lệ Runtime khi thực thi mã này.[2]

* 1. ***Không thể hoạt động trên đối tượng miền (đối tượng miền bị mất)***

Một khi bạn đã chuyển đổi một đối tượng miền thành dataframe, bạn không thể tạo lại nó từ chính nó.[2]

# Tài liệu tham khảo

1. <URL: <https://blog.vietnamlab.vn/xu-ly-du-lieu-voi-spark-dataframe/>>
2. <URL: <https://qastack.vn/programming/31508083/difference-between-dataframe-dataset-and-rdd-in-spark>>
3. <URL: <https://helpex.vn/article/huong-dan-pyspark-dataframe-gioi-thieu-ve-dataframes-5c6b21e6ae03f628d053c29e>>