GENERAL KNOWLEDGE OF
MINING OF MASSIVE DATASETS

MINING OF MASSIVE DATASETS

DANG HOANG DAI NGHIA

TON DUC THANG
UNIVERSITY

MỤC LỤC

Spark

I. Khái niệm: [1]

Apache Spark cho phép xây dựng các mô hình dự đoán nhanh chóng với việc tính toán được thực hiện trên một nhóm các máy tính, có có thể tính toán cùng lúc trên toàn bộ tập dữ liệu mà không cần phải trích xuất mẫu tính toán thử nghiệm. Tốc độ xử lý của Spark có được do việc tính toán được thực hiện cùng lúc trên nhiều máy khác nhau. Đồng thời việc tính toán được thực hiện ở bộ nhớ trong (in-memories) hay thực hiện hoàn toàn trên RAM.

Spark cho phép xử lý dữ liệu theo thời gian thực, vừa nhận dữ liệu từ các nguồn khác nhau đồng thời thực hiện ngay việc xử lý trên dữ liệu vừa nhận được (Spark Streaming).

Spark không có hệ thống file của riêng mình, nó sử dụng hệ thống file khác như: HDFS, Cassandra, S3,…. Spark hỗ trợ nhiều kiểu định dạng file khác nhau (text, csv, json…) đồng thời nó hoàn toàn không phụ thuộc vào bất cứ một hệ thống file nào.

II. Thành phần của Spark: [1]

Apache Spark gồm có 5 thành phần chính : Spark Core, Spark Streaming, Spark SQL, MLlib và GraphX, trong đó:

Thành phần trung của Spark là Spark Core: cung cấp những chức năng cơ bản nhất của Spark như lập lịch cho các tác vụ, quản lý bộ nhớ, fault recovery, tương tác với các hệ thống lưu trữ…Đặc biệt, Spark Core cung cấp API để định nghĩa RDD (Resilient Distributed DataSet) là tập hợp của các item được phân tán trên các node của cluster và có thể được xử lý song song. [2]

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.

• 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.

• Spark Streaming cung cấp API để dễ dàng xử lý dữ liệu stream,

• MLlib Cung cấp rất nhiều thuật toán của học máy như: classification, regression, clustering, collaborative filtering…

• GraphX là thư viện để xử lý đồ thị.

Trong các thư viện mà Spark cung cấp thì có 69% người dùng Spark SQL, 62% sử dụng DataFrames, Spark Streaming và MLlib + GraphX là 58%

III. Những đặc điểm nổi bật: [1]

Xử lý dữ liệu: Spark xử lý dữ liệu theo lô và thời gian thực

Tính tương thích: Có thể tích hợp với tất cả các nguồn dữ liệu và định dạng tệp được hỗ trợ bởi cụm Hadoop

Hỗ trợ ngôn ngữ: hỗ trợ Java, Scala, Python và R

Phân tích thời gian thực

Mục tiêu sử dụng:

• Xử lý dữ liệu nhanh và tương tác

• Xử lý đồ thị

• Công việc lặp đi lặp lại

• Xử lý thời gian thực

• joining Dataset

• Machine Learning

• Apache Spark là Framework thực thi dữ liệu dựa trên Hadoop HDFS. Apache Spark không thay thế cho Hadoop nhưng nó là một framework ứng dụng. Apache Spark tuy ra đời sau nhưng được nhiều người biết đến hơn Apache Hadoop vì khả năng xử lý hàng loạt và thời gian thực.

MapReduce

I. Khái niệm: [2]

Mapreduce có thể hiểu là 1 phương thức thực thi để giúp các ứng dụng có thể xử lý nhanh 1 lượng dữ liệu lớn. Các dữ liệu này được đặt tại các máy tính phân tán.Các máy tính này sẽ hoạt động song song độc lập với nhau.Điều này làm rút ngắn thời gian xử lý toàn bộ dữ liệu.

Một đặc điểm đáng chú ý của Mapreduce là dữ liệu đầu vào có thể là dữ liệu có cấu trúc (dữ liệu lưu trữ dạng bảng quan hệ 2 chiều) hoặc dữ liệu không cấu trúc (dữ liệu dạng tập tin hệ thống).

Các máy tính lưu trữ các dữ liệu phân tán trong quá trình thực thi được gọi là các nút (nodes) của hệ thống.Nếu các máy tính này cùng sử dụng chung trên 1 phần cứng thì chúng được gọi là 1 cụm (Cluster).Nếu các máy này hoatj động riêng rẽ trên các phần cứng khác nhau thì chúng được gọi là 1 lưới (Grid).

II. Các hàm chính của MapReduce: [3]

MapReduce có 2 hàm chính là Map() và Reduce(), đây là 2 hàm đã được định nghĩa bởi người dùng và nó cũng chính là 2 giai đoạn liên tiếp trong quá trình xử lý dữ liệu của MapReduce. Nhiệm vụ cụ thể của từng hàm như sau:

• Hàm Map(): có nhiệm vụ nhận Input cho các cặp giá trị/ khóa và output chính là tập những cặp giá trị/khóa trung gian. Sau đó, chỉ cần ghi xuống đĩa cứng và tiến hành thông báo cho các hàm Reduce() để trực tiếp nhận dữ liệu.

• Hàm Reduce(): có nhiệm vụ tiếp nhận từ khóa trung gian và những giá trị tương ứng với lượng từ khóa đó. Sau đó, tiến hành ghép chúng lại để có thể tạo thành một tập khóa khác nhau. Các cặp khóa/giá trị này thường sẽ thông qua một con trỏ vị trí để đưa vào các hàm reduce. Quá trình này sẽ giúp cho lập trình viên quản lý dễ dàng hơn một lượng danh sách cũng như phân bổ giá trị sao cho phù hợp nhất với bộ nhớ hệ thống.

• Ở giữa Map và Reduce thì còn 1 bước trung gian đó chính là Shuffle. Sau khi Map hoàn thành xong công việc của mình thì Shuffle sẽ làm nhiệm vụ chính là thu thập cũng như tổng hợp từ khóa/giá trị trung gian đã được map sinh ra trước đó rồi chuyển qua cho Reduce tiếp tục xử lý.

III. Những đặc điểm nổi bật: [3]

Mapreduce được ưa chuộng sử dụng như vậy bởi nó sở hữu nhiều ưu điểm vượt trội như sau:

• MapReduce có khả năng xử lý dễ dàng mọi bài toán có lượng dữ liệu lớn nhờ khả năng tác vụ phân tích và tính toán phức tạp. Nó có thể xử lý nhanh chóng cho ra kết quả dễ dàng chỉ trong khoảng thời gian ngắn.

• Mapreduce có khả năng chạy song song trên các máy có sự phân tán khác nhau. Với khả năng hoạt động độc lập kết hợp phân tán, xử lý các lỗi kỹ thuật để mang lại nhiều hiệu quả cho toàn hệ thống.

• MapRedue có khả năng thực hiện trên nhiều nguồn ngôn ngữ lập trình khác nhau như: Java, C/ C++, Python, Perl, Ruby,… tương ứng với nó là những thư viện hỗ trợ.

• Như bạn đã biết, mã độc trên internet ngày càng nhiều hơn nên việc xử lý những đoạn mã độc này cũng trở nên rất phức tạp và tốn kém nhiều thời gian. Chính vì vậy, các ứng dụng MapReduce dần hướng đến quan tâm nhiều hơn cho việc phát hiện các mã độc để có thể xử lý chúng. Nhờ vậy, hệ thống mới có thể vận hành trơn tru và được bảo mật nhất.

SO SÁNH SPARK VÀ MAP REDUCE [4]

	Spark	Map Reduce
Xử lý dữ liệu	Spark tăng tốc xử lý hàng loạt thông qua tính toán trong bộ nhớ và tối ưu hóa xử lý.	Map Reduce lưu trữ dữ liệu trên đĩa, có nghĩa là nó có thể xử lý các bộ dữ liệu lớn.
Tính tương thích	Có thể tích hợp với tất cả các nguồn dữ liệu và định dạng tệp được hỗ trợ bởi cụm Hadoop.	tương thích chủ yếu nhiều các data sources và định dạng tệp.
Phân tích thời gian thực	Spark có thể xử lý dữ liệu thời gian thực, tức là dữ liệu đến từ các luồng sự kiện thời gian thực với tốc độ hàng triệu sự kiện mỗi giây, chẳng hạn như dữ liệu Twitter và Facebook. Sức mạnh của Spark nằm ở khả năng xử lý luồng trực tiếp hiệu quả.	MapReduce thất bại khi xử lý dữ liệu thời gian thực, vì nó được thiết kế để thực hiện xử lý hàng loạt trên lượng dữ liệu khổng lồ.
Hỗ trợ ngôn ngữ	Spark hỗ trợ Java, Scala, Python và R.	MapRedue có khả năng thực hiện trên nhiều nguồn ngôn ngữ lập trình khác nhau như: Java, C/ C++, Python, Perl, Ruby,… tương ứng với nó là những thư viện hỗ trợ.
Xử lý đồ thị	Spark đi kèm với một thư viện tính toán đồ thị có tên là GraphX để làm cho mọi thứ trở nên đơn giản. Tính toán trong bộ nhớ kết hợp với hỗ trợ đồ thị dựng sẵn cho phép thuật toán thực hiện tốt hơn nhiều so với các chương trình MapReduce truyền thống.	MapReduce đọc dữ liệu từ đĩa và sau một lần lặp cụ thể, sẽ gửi kết quả đến HDFS, rồi lại đọc dữ liệu từ HDFS cho lần lặp tiếp theo. Quá trình này làm tăng độ trễ và làm cho xử lý đồ thị chậm.

Bảo mật	Bảo mật của Spark hiện đang ở giai đoạn đầu, chỉ cung cấp hỗ trợ xác thực thông qua bí mật chung (xác thực mật khẩu).	Map Reduce có các tính năng bảo mật tốt hơn Spark. Hadoop MapReduce cũng có thể tích hợp với các dự án bảo mật của Hadoop, như Knox Gateway và Sentry.

TÀI LIỆU THAM KHẢO

[1]: https://viblo.asia/p/tim-hieu-ve-apache-spark-ByEZkQQW5Q0

[2]: https://sites.google.com/site/chapterhut/hoc-tap/mon-hoc/map-reduce

[3]: https://blog.itnavi.com.vn/mapreduce-nhung-uu-diem-va-cach-thuc-hoat-dong-cua-nen-tang-nay/

[4]: https://helpex.vn/article/hadoop-mapreduce-so-voi-apache-spark-5c6b19eaae03f628d053bd1a

[]: https://cloudfun.vn/threads/phan-biet-apache-hadoop-va-apache-spark.94/