**Spark**

Apache Spark là một open source cluster computing framework 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.

Các thành phần của Apache Spark:

+ Thành phần cốt lõi 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.

+ Spark SQL là một thành phần nằm trên Spark Core 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 (Machine Learning Library) là một nền tảng học máy phân tán bên trên Spark do kiến trúc phân tán dựa trên bộ nhớ. Nó cung cấp rất nhiều thuật toán của học máy như: classification, regression, clustering, collaborative filtering…

+ GraphX là thư viện để xử lý đồ thị. Nó cung cấp các API để diễn tảcác tính toán trong đồ thị bằng cách sử dụng Pregel Api.

Những tính năng nổi bật của 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.
* Stop job đang chạy bằng cách stop spark context.
* Bỏ bước upload gói jar lúc start job làm cho job được start nhanh hơn.
* Cung cấp hai cơ chế chạy job đồng bộ và bất đồng bộ.
* Cho phép cache RDD theo tên , tăng tính chia sẻ và sử dụng lại RDD giữa các job.
* Hỗ trợ viết spark job bằng cú pháp SQL.
* Dễ dàng tích hợp với các công cụ báo cáo như: Business Intelligence, Analytics, Data Integration Tools.

Về quản lý bộ nhớ, Spark giải quyết các vấn đề vấn đề xung quanh định nghĩa Resilient Distributed Datasets (RDDs). RDDs hỗ trợ hai kiểu thao tác thao tác: transformations và action. Thao tác chuyển đổi(tranformation) tạo ra dataset từ dữ liệu có sẵn. Thao tác actions trả về giá trị cho chương trình điều khiển (driver program) sau khi thực hiện tính toán trên dataset.

**Mapreduce**

Mapreduce là mô hình được thiết kế độc quyền bởi Google, nó có khả năng lập trình xử lý các tập dữ liệu lớn song song và phân tán thuật toán trên 1 cụm máy tính. Mapreduce trở thành một trong những thành ngữ tổng quát hóa trong thời gian gần đây. Mapreduce sẽ bao gồm những thủ tục sau: thủ tục 1 Map() và 1 Reduce(). Thủ tục Map() bao gồm lọc (filter) và phân loại (sort) trên dữ liệu khi thủ tục khi thủ tục Reduce() thực hiện quá trình tổng hợp dữ liệu. Đây là mô hình dựa vào các khái niệm biển đối của bản đồ và reduce những chức năng lập trình theo hướng chức năng. Thư viện của thủ tục Map() và Reduce() sẽ được viết bằng nhiều loại ngôn ngữ khác nhau. Thủ tục được cài đặt miễn phí và được sử dụng phổ biến nhất là là Apache Hadoop.

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.

Các ưu điểm nổi bật của Mapreduce:

* 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.
* Mapreduce 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.

Nguyên tắc hoạt động của Mapreduce: hoạt động dựa vào nguyên tắc chính là “Chia để trị”, như sau:

* Phân chia các dữ liệu cần xử lý thành nhiều phần nhỏ trước khi thực hiện.
* Xử lý các vấn đề nhỏ theo phương thức song song trên các máy tính rồi phân tán hoạt động theo hướng độc lập.
* Tiến hành tổng hợp những kết quả thu được để đề ra được kết quả sau cùng.

Các bước hoạt động của Mapreduce:

* Bước 1: Tiến hành chuẩn bị các dữ liệu đầu vào để cho Map() có thể xử lý.
* Bước 2: Lập trình viên thực thi các mã Map() để xử lý.
* Bước 3: Tiến hành trộn lẫn các dữ liệu được xuất ra bởi Map() vào trong Reduce Processor
* Bước 4: Tiến hành thực thi tiếp mã Reduce() để có thể xử lý tiếp các dữ liệu cần thiết.
* Bước 5: Thực hiện tạo các dữ liệu xuất ra cuối cùng.

Luồng dữ liệu nền tảng của Mapreduce:

* Input Reader
* Map Function
* Partition Function
* Compare Function
* Reduce Function
* Output Writer

Mapreduce được ứng dụng cho việc thống kê hàng loạt những số liệu cụ thể như:

* Thực hiện thống kê cho các từ khóa được xuất hiện ở trong các tài liệu, bài viết, văn bản hoặc được cập nhật trên hệ thống fanpage, website,…
* Khi số lượng các bài viết đã được thống kê thì tài liệu sẽ có chứa các từ khóa đó.
* Sẽ có thể thống kê được những câu lệnh match, pattern bên trong các tài liệu đó
* Khi thống kê được số lượng các URLs có xuất hiện bên trong một webpages.
* Sẽ thống kê được các lượt truy cập của khách hàng sao cho nó có thể tương ứng với các URLs.
* Sẽ thống kê được tất cả từ khóa có trên website, hostname,…