**Spark và Mapreduce**

1. **Spark**

**Tổng quan về Apache 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 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 còn do việc tính toán được thực hiện ở bộ nhớ trong hoặc trên RAM.

Với các tính năng nổi bật như xử lý nhanh, linh hoạt và thân thiện với người dùng, Apache Spark nhanh chóng trở thành nền tảng hàng đầu giúp phân phối và xử lý dữ liệu lớn trong thế giới kỹ thuật số đang phát triển mạnh đòi hỏi những sự tính toán khổng lồ.

Spark có thể được triển khai bằng nhiều ngôn ngữ lập trình như: Java, Scala, Python và R, đồng thời hỗ trợ SQL. Hiện nay, nó được ứng dụng bởi các ngân hàng, công ty viễn thông, công ty trò chơi và những gã công nghệ lớn như Apple, Facebook, Twitter, IBM và Microsoft, …

**Thành phần của Apache Spark**

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

Spark Core là nền tảng cho các thành phần còn lại và các thành phần này muốn khởi chạy được thì đều phải thông qua Spark Core. Nó đảm nhận vai trò thực hiện công việc tính toán và xử lý bộ nhớ trong đồng thời nó cũng tham chiếu các dữ liệu được lưu trữ tại các hệ thống lưu trữ bên ngoài.

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.

**Quản lý bộ nhớ của Apache Spark**

Ý tưởng chính của Spark là Resilient Distributed Datasets (RDD); nó hỗ trợ tính toán xử lý trong bộ nhớ. Điều này có nghĩa, nó lưu trữ trạng thái của bộ nhớ dưới dạng một đối tượng trên các công việc và đối tượng có thể chia sẻ giữa các công việc đó.

RDDs có thể chứa bất kỳ kiểu dữ liệu nào của Python, Java, hoặc đối tượng Scala, bao gồm các kiểu dữ liệu do người dùng định nghĩa. Thông thường, RDD chỉ cho phép đọc, phân mục tập hợp của các bản ghi.

Có hai cách để tạo RDDs:

* Tạo từ một tập hợp dữ liệu có sẵn trong ngôn ngữ sử dụng như Java, Python, Scala.
* Lấy từ dataset hệ thống lưu trữ bên ngoài như HDFS, Hbase hoặc các cơ sở dữ liệu quan hệ.

1. **Mapreduce**

MapReduce được thiết kế như 1 mô hình lập trình xử lý tập dữ liệu lớn song song, thuật toán được phân tán trên 1 cụm.

MapReduce gồm các thủ tục: 1 Map() và 1 Reduce().

Thủ tục Map() lọc và phân loại dữ liệu .

Thủ tục Reduce() thực hiện tổng hợp dữ liệu.

Thư viện thủ tục Map() và Reduce() được viết bằng nhiều ngôn ngữ.

Nền tảng MapReduce có 5 bước khác nhau:

* Chuẩn bị dữ liệu đầu vào cho Map()
* Thực thi mã Map() được cung cấp bởi người dùng
* Trộn dữ liệu xuất của Map vào Reduce Processor
* Thực thi mã Reduce() được cung cấp bởi người dùng
* Tạo dữ liệu xuất cuối cùng