**SPARK**

Spark cho phép xây dựng các mô hình dự đoán nhanh chóng và được thực hiện bởi nhiều máy tính trong cùng một lúc trên một cơ sở dữ liệu.

Các thành phần của Spark:

* Spark core: giúp phân chia công việc cho các máy tính, quản lý bộ nhớ, giao tiếp với hệ thống lưu trữ,… Đồng thời Spark Core còn cung cấp RDD (Resilent Distributed Dataset) tập hợp các item được phân bổ trên các node của cluster và có thể xử lý song song với nhau.
* Spark SQL: cho phép dữ liệu được truy vấn qua các dòng lệnh SQL.
* Spark Streaming: xử lý dữ liệu stream.
* Mllib: cung cấp các thuật toán như classification, regression,…
* GraphX: giúp vẽ đồ thị.

Ngôn ngữ sử dụng Spark gồm nhiều ngôn ngữ. Năm 2014, 84% người sử dụng Scala, và Java, Python chiếm 38%. Tuy nhiên đến năm 2015 thì Spark có hỗ trợ thêm ngôn ngữ R chiếm 18% người dùng còn Python tăng lên 58%.

Các công ty đang sử dụng Spark là Yahoo, Twitter, Ebay, Netflix,…

**MAPREDUCE**

MapReduce là mô hình được thiết kế bởi Google, nó giúp xử lý các tập dữ liệu lớn một cách song song trên 1 cụm các máy tính. Và nó gồm 2 thao tác chính là Map và Reduce.

* Map: giúp chọn lọc, phân loại dữ liệu trong lúc xử lý. Và nó hoạt động chi tiết như sau: nó sẽ giúp nhận input cho các giá trị có dạng key-value và giúp xuất ra các output cũng có dạng key-value, đồng thời thông báo cho Reduce tiếp nhận giá trị đó để xử lý ra kết quả.
* Reduce: tổng hợp dữ liệu cuối cùng.
* Ở giữa Map và Reduce còn có Shuffle. Sau khi Map thực hiện nhiệm vụ xong thì Shuffle sẽ giúp tổng hợp lại các khóa giá trị để chuyển cho Reduce.

MapReduce được thực hiện trên nhiều ngôn ngữ khác nhau như Java, C/C++, Python,…Và ưu điểm của MapReduce ở thời đại Internet hiện nay là xử lý những đoạn mã độc trở nên nhanh hơn và đở tốn kém thời gian hơn.

Nguyên tắc hoạt động:

MapReduce hoạt động theo nguyên tắc chia để trị. Nghĩa là nó sẽ phân dữ liệu thành các phần nhỏ trước khi thực hiện, sau đó nó sẽ xử lý các phần nhỏ đó trên nhiều máy tính một cách song song và cuối cùng là tổng hợp lại kết quả.

Các bước hoạt động:

* Bước 1: Truyền dữ liệu đầu vào để cho Map() 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.