**Tên: Nguyễn Anh Kiệt**

**MSSV: 51703119**

**Tổng qua về Mapreduce**

**Khái niệm** :  
Mapreduce là một mô hình lập trình, thực hiện quá tình xử lý tập dữ liệu lớn. Mapreduce gồm 2 pha : map và reduce.  
Hàm Map : Các xử lý một cặp (key, value) để sinh ra một cặp (keyI, valueI) - key và value trung gian. Dữ liệu này input vào hàm Reduce.  
Hàm Reduce : Tiếp nhận các (keyI, valueI) và trộn các cặp (keyI, valueI) trung gian , lấy ra các valueI có cùng keyI.

Việc của lập trình viên là quan tâm tới 2 hàm Map và Reduce. Còn các vấn đề khác như : phân chia các dữ liệu đầu vào, lịch trình thực thi các machines, handling các machines failure, quản lý việc giao tiếp giữa các machines là việc của hệ thống run-time.  
=> Lập trình viên có thể không có kinh nghiệm về hệ thống song song và phân tán vẫn dễ dàng vận hành một hệ thống phân tán lớn.  
Áp dụng mô hình MapReduce chạy trên lượng lớn các machine cỡ hàng ngàn machine và data lên đến mức Terabytes.

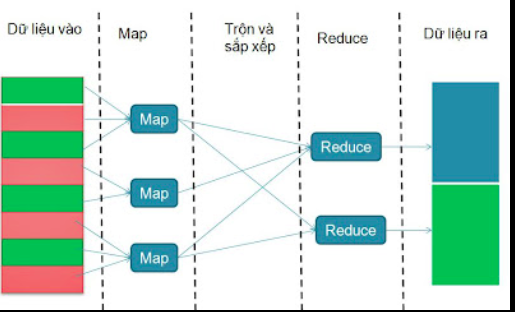
Các job sau dễ dàng sử dụng Mapreduce:

* Thống kê số từ khóa xuất hiện trong các documents.
* Thống kê số documents có chứa từ khóa.
* Thống kê số câu match với pattern trong các documents.
* Thống kê số URLs xuất hiện trong các web pages.
* Thống kê số lượt truy cập các URLs.
* Thống kê số từ khóa trên các hostnames.
* Distributed Sort.

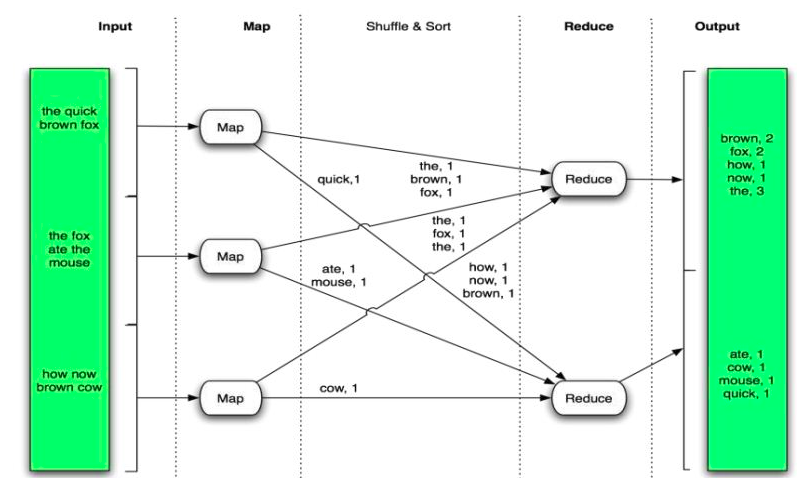
**Hoạt động** :

* Ý tưởng
  + Chia vấn đề cần xử lý thành các phần nhỏ để xử lý.
  + Xử lý các phần nhỏ đó một cách song song và độc lập trên các máy tính phân tán.
  + Tổng hợp các kết quả thu được để dưa ra kết quả cuối cùng.
* Hoạt động của MapReduce có thể được tóm tắt như sau:
  + Đọc dữ liệu đầu vào
  + Xử lý dữ liệu đầu vào (thực hiện hàm map)
  + Sắp xếp và trộn các kết quả thu được từ các máy tính phân tán thích hợp nhất.
  + Tổng hợp các kết quả trung gian thu được ( thực hiện hàm reduce)
  + Đưa ra kết quả cuối cùng.

S**ơ đồ** :

[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/m%C3%B4%20h%C3%ACnh%20ho%E1%BA%A1t%20%C4%91%E1%BB%99ng.png_5a3zte8t56)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/m%C3%B4%20h%C3%ACnh%20ho%E1%BA%A1t%20%C4%91%E1%BB%99ng.png_5a3zte8t56)

Ví dụ với bài toán wordcount

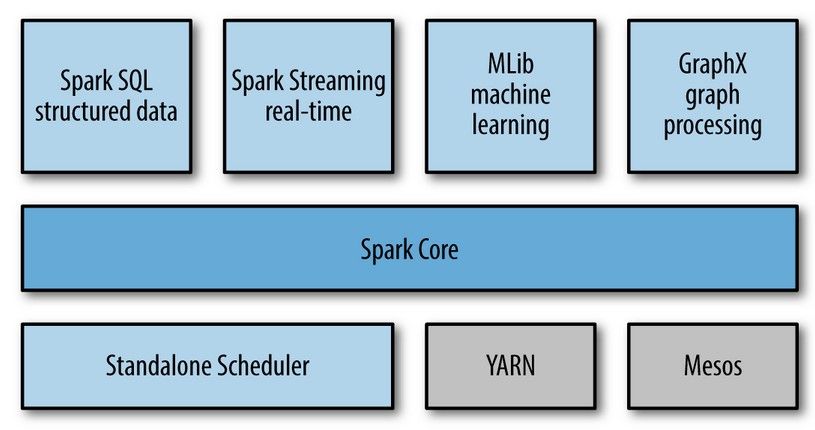
[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/wordcount.png_yvridty0yq)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/wordcount.png_yvridty0yq)

**Tổng quan về Apache Spark**

**Tổng quan.**

Apache Spark là một open source cluster computing framework được phát triển sơ khởi vào năm 2009 bởi AMPLab tại đại học California. Sau này, Spark đã được trao cho Apache Software Foundation vào năm 2013 và được phát triển cho đến nay. Nó 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.

**Thành phần của Apache Spark**

Matei Zaharia, cha đẻ của Spark, sử dụng Hadoop từ những ngày đầu. Đến năm 2009 ông viết Apache Spark để giải quyết những bài toán học máy ở đại học UC Berkely vì Hadoop MapReduce hoạt động không hiệu quả cho những bài toán này. Rất sớm sau đó ông nhận ra rằng Spark không chỉ hữu ích cho học máy mà còn cho cả việc xử lý luồng dữ liệu hoàn chỉnh.

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.

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%

LINK GITHUB: [git@github.com:kietnguyen305/Kiem-thu-phan-mem.git](mailto:git@github.com:kietnguyen305/Kiem-thu-phan-mem.git)

Link google colab: https://colab.research.google.com/drive/1pBvWJ4rm6ZNpSap6AhaYy7AjZxxtjuyf