**HOMEWORK 1: Tìm hiểu về Spark**

**1/ Spark RDD**

**Khái niệm:**

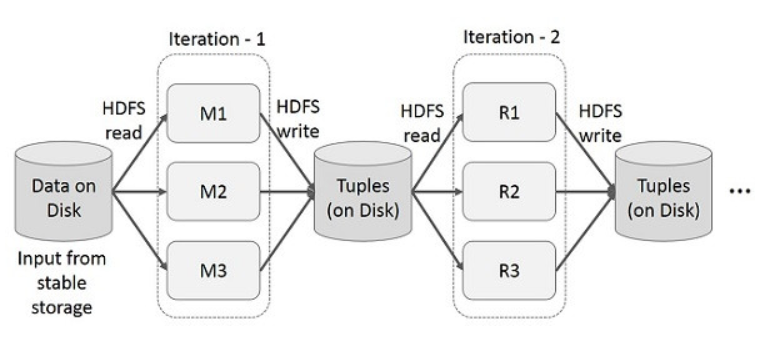
Ở cấp độ cao, mọi ứng dụng Spark bao gồm một chương trình trình điều khiển chạy chức năng chính của người dùng và thực hiện các hoạt động song song khác nhau trên một cụm. Tính trừu tượng cơ bản mà Spark cung cấp là tập dữ liệu phân tán có khả năng phục hồi (RDD), là tập hợp các phần tử được phân vùng trên các nút của cụm có thể hoạt động song song.

**Thực thi trên MapReduce:**

- MapReduce dùng để xử lý và tạo các bộ dữ liệu lớn với thuật toán xử lý phân tán song song trên một cụm. Nó cho phép người dùng viết các tính toán song song, sử dụng một tập hợp các toán tử cấp cao, mà không phải lo lắng về xử lý/phân phối công việc và khả năng chịu lỗi.

- Hầu hết các framework hiện tại, cách duy nhất để sử dụng lại dữ liệu giữa các tính toán (Ví dụ: giữa hai công việc MapReduce) là ghi nó vào storage (Ví dụ: HDFS)

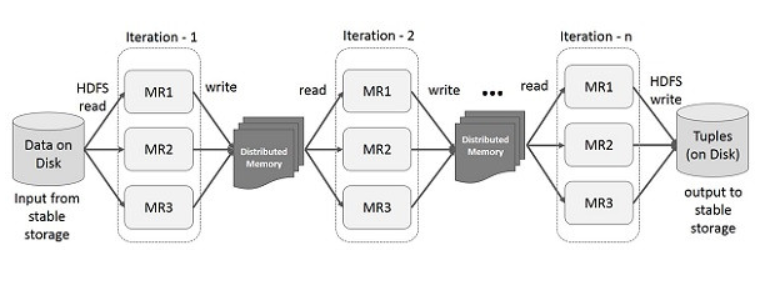
- Việc chia sẻ dữ liệu chậm trong MapReduce là do sao chép tuần tự và tốc độ I/O của ổ đĩa.



Hình 1. Minh họa việc thực thi trên MapReduce

**Thực thi trên Spark RDD:**

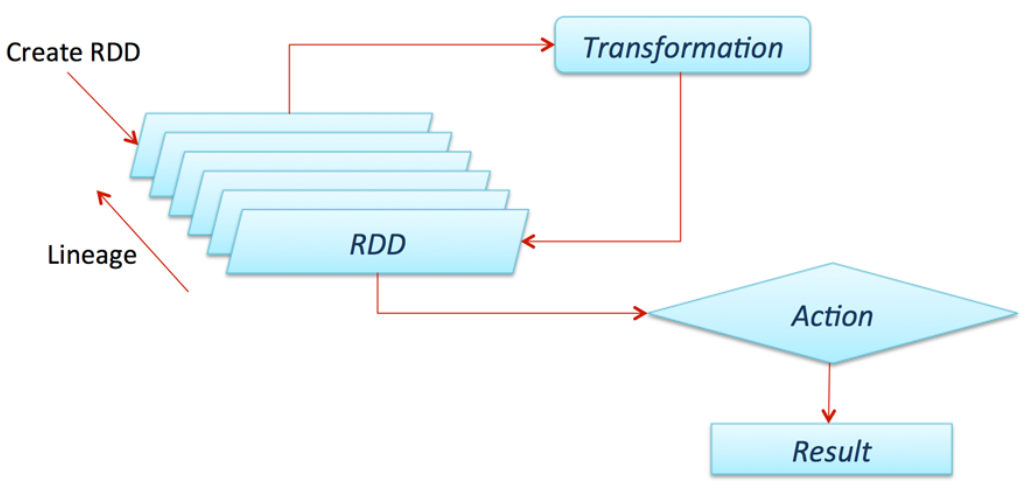
Để khắc phục vấn đề về MapRedure, Spark RDD ra đời để hỗ trợ tính toán xử lý trong bộ nhớ. Việc xử lý dữ liệu trong bộ nhớ nhanh hơn 10 đến 100 lần so với network và disk.



Hình 2. Minh họa việc thực thi trên Spark RDD

**Các transformation và action với RDD:**

Transformation xử lý các thao tác lazily và Action xử lý thao tác cần xử lý tức thời.



Hình 3. Flow của transformation và action trong RDD

* **Transformation:** biến đổi từ RDD này sang RDD khác.

- map(func): Trả về một tập dữ liệu phân tán mới được hình thành bằng cách chuyển từng phần tử của nguồn thông qua một hàm.

- filter(func): Trả về một tập dữ liệu mới là các phần tử được lọc ra từ func trả về true.

- distinct(): Loại bỏ trùng lắp trong RDD.

- flatMap(func): Tương tự như hàm map. Nhưng sau đó nó sẽ làm phẳng kết quả.

- sortBy(keyfunc): Sắp xếp RDD theo keyfunc.

- reduceByKey(func): Hợp nhất các giá trị cho mỗi khóa.

* **Action:**

- collect(): Trả về tất cả các phần tử của tập dữ liệu dưới dạng một mảng tại driver.

- count(): Trả về số phần tử trong tập dữ liệu.

- first(): Lấy giá trị đầu tiên của tập dữ liệu.

- reduce(func): Thực hiện hàm reduce trên RDD để thu về 1 giá trị duy nhất.

- min(), max(): Tìm giá trị nhỏ nhất và lớn nhất trong RDD.

- take(n): Trả về một mảng có n phần tử đầu tiên của tập dữ liệu.

**TÀI LIỆU THAM KHẢO**

1. Apache Spark RDD, <[Apache Spark RDD | Apache Spark | Laptrinh.vn](https://laptrinh.vn/books/apache-spark/page/apache-spark-rdd)>