# **GIỚI THIỆU VỀ APACHE SPARK VÀ MAPREDUCE**

* **Apache Spark:** là một cụm tính toán các thư viện mã nguồn mở được phát triển bởi AMPLab tại đại học Califonia sau này được chuyển sáng cho Apache Software Foundation và được phát triển cho đén ngày nay

Spark cho phép xây dựng mô phỏng tính toán trên 1 cụm máy tính, có thể tính toán cùng lúc trên những máy tính đó và cùng trên một tập dữ liệu mà không phải trích xuất mẫu toán thử nghiệm

tốc độ xử lý của Spark có được do việc tính toán được thực hiện trên nhiều máy tính khác nhau. Đồng thời việc tính toán được thực hiện ở bộ nhớ trong hoặc có thể thực hiện trên Ram

Thành phần trung gian của Spark là Spark core: cung cấp những chức năng cơ bản như lưu lại lịch sử của các tác vụ, quản lý bộ nhớ,fault recovery, tưởng tác với các hệ thống lưu trữ và 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 xử lý song song. Spark core có thể hoạt động tốt với thư viện giành cho SQL, Machine learning, stream processing. Hoạt động với các API cho các ngôn ngữ như Java, Scala, Python, và R

* **MapReduce:** được tự cho là “ Một framework phần mềm tạo ra để dễ dàng viết các ứng dụng xử lý lượng lớn dữ liệu(tập dữ liệu nhiều terabyte) song song trên các cluster lớn(lên đến hàng ngàn nodes) của phần cứng theo cách đáng tin cậy,

MapReduce: bao gồm 2 thành phần tuần tự thực hiện các tác vụ: Map (được hiểu là ánh xạ) và Reduce(rút gọn). Bộ lọc ánh xạ và sắp xếp dữ liệu trọng khi chuyển đổi chúng thành các cặp khóa và giá trị, Sau khi kết thúc quá trình ánh xạ sẽ thưc hiện tiếp tác vụ là rút gọn.Rút gòn là việc lấy đầu vào từ ánh xạ sau đó giảm kích thước của dữ liệu nhận vào bằng cách thực hiện một số cách để thu gọn tập dữ liệu lại. MapReduce có thể tăng tốc đáng kể các tác vụ dữ liệu lớn bằng cách chia nhỏ dữ liệu lớn rồi xử lý trên từng phần nhỏ đó 1 cách song song

# **SO SÁNH**

Sự Khác nhau giữa Spark và MapReduce:

* Hiệu suất
* Dễ sử dụng
* Xử lí dữ liệu
* Bảo mật

Các điểm tương đồng giữa Spark và MapReduce

* Giá cả
* Khả năng tương thích
* Rủi ro thất bại

## **KHÁC NHAU**

|  |  |  |
| --- | --- | --- |
|  | Apache Spark | MapReduce |
| Hiệu Suất | Apache Spark xử lý dữ liệu trong bộ nhớ truy cập ngẫu nhiên (RAM),  Spark cần rất nhiều bộ nhớ. Giống như cơ sở dữ liệu tiêu chuẩn, Spark tải một quy trình vào bộ nhớ và giữ nó ở đó cho đến khi có thông báo mới để lưu vào bộ nhớ đệm.Nếu bạn chạy Spark trên Hadoop YARN với các dịch vụ đòi hỏi tài nguyên khác hoặc nếu dữ liệu quá lớn để vừa hoàn toàn vào bộ nhớ thì Spark có thể bị suy giảm hiệu suất nghiêm trọng. | Hadoop MapReduce duy trì dữ liệu quay trở lại đĩa sau một bản đồ hoặc giảm hoạt động  MapReduce hủy các quy trình của nó ngay sau khi hoàn thành công việc, vì vậy nó có thể dễ dàng chạy cùng với các dịch vụ khác với sự khác biệt nhỏ về hiệu suất. |
| Dễ sử dụng | Spark có các API được tạo sẵn cho Java, Scala và Python, và cũng bao gồm Spark SQL (trước đây gọi là Shark) cho người hiểu biết về SQL. Nhờ các khối xây dựng đơn giản của Spark, thật dễ dàng để viết các hàm do người dùng xác định. Spark thậm chí còn bao gồm một chế độ tương tác để chạy các lệnh với phản hồi ngay lập tức. | MapReduce được viết bằng Java và nổi tiếng là rất khó lập trình. Apache Pig làm cho nó dễ dàng hơn (mặc dù nó đòi hỏi một thời gian để học cú pháp), trong khi Apache Hive bổ sung khả năng tương thích SQL cho đĩa. Một số công cụ Hadoop cũng có thể chạy các công việc MapReduce mà không cần lập trình. Ví dụ: Xplenty là một dịch vụ tích hợp dữ liệu được xây dựng trên Hadoop và cũng không yêu cầu bất kỳ lập trình hoặc triển khai nào. |
| Xử Lí Dữ Liệu | Spark có thể làm được nhiều việc hơn là xử lý dữ liệu đơn thuần: nó cũng có thể xử lý đồ thị và bao gồm thư viện máy học MLlib. Nhờ hiệu suất cao, Spark có thể xử lý theo thời gian thực cũng như xử lý hàng loạt. Spark cung cấp nền tảng "một kích thước phù hợp với tất cả" mà bạn có thể sử dụng thay vì chia nhỏ các nhiệm vụ trên các nền tảng khác nhau, điều này làm tăng thêm sự phức tạp về CNTT của bạn. | Hadoop MapReduce rất tuyệt vời để xử lý hàng loạt. Nếu muốn có tùy chọn thời gian thực, bạn sẽ cần sử dụng một nền tảng khác như Impala hoặc Apache Storm và để xử lý đồ thị, bạn có thể sử dụng Apache Giraph. MapReduce từng có Apache Mahout cho máy học, nhưng kể từ đó, nó đã được chuyển sang sử dụng Spark và H2O. |
| Bảo Mật | Về mặt bảo mật, Spark kém tiên tiến hơn khi so sánh với MapReduce. Trên thực tế, bảo mật trong Spark được đặt thành tắt theo mặc định, điều này có thể khiến bạn dễ bị tấn công.  Xác thực trong Spark được hỗ trợ cho các kênh RPC thông qua một bí mật được chia sẻ. Spark bao gồm ghi nhật ký sự kiện như một tính năng và giao diện người dùng Web có thể được bảo mật thông qua các bộ lọc javax servlet. Ngoài ra, vì Spark có thể chạy trên YARN và sử dụng HDFS, nó cũng có thể tận hưởng xác thực Kerberos, quyền đối với tệp HDFS và mã hóa giữa các nút. | Hadoop MapReduce có thể tận hưởng tất cả các lợi ích bảo mật của Hadoop và tích hợp với các dự án bảo mật Hadoop, như Knox Gateway và Apache Sentry. Dự án Rhino, nhằm mục đích cải thiện bảo mật của Hadoop, chỉ đề cập đến Spark liên quan đến việc thêm hỗ trợ Sentry. Nếu không, các nhà phát triển Spark sẽ phải tự cải thiện bảo mật Spark. |

## **GIỐNG NHAU**

Spark và MapReduce là các giải pháp mã nguồn mở, nhưng bạn vẫn cần chi tiền cho máy móc và nhân viên

Apache Spark có thể chạy như một ứng dụng độc lập, trên Hadoop YARN hoặc Apache Mesos tại chỗ hoặc trên đám mây. Spark hỗ trợ các nguồn dữ liệu triển khai Hadoop InputFormat, vì vậy nó có thể tích hợp với tất cả các nguồn dữ liệu và định dạng tệp giống nhau mà Hadoop hỗ trợ. Spark cũng hoạt động với các công cụ thông minh kinh doanh thông qua JDBC và ODBC.

Spark có các lần thử lại cho mỗi tác vụ và thực thi suy đoán, giống như MapReduce. Tuy nhiên, MapReduce có một chút lợi thế ở đây vì nó dựa trên ổ cứng, thay vì RAM. Nếu một quá trình MapReduce gặp sự cố khi đang thực thi, nó có thể tiếp tục ở nơi nó đã dừng lại, trong khi Spark sẽ phải bắt đầu xử lý từ đầu.