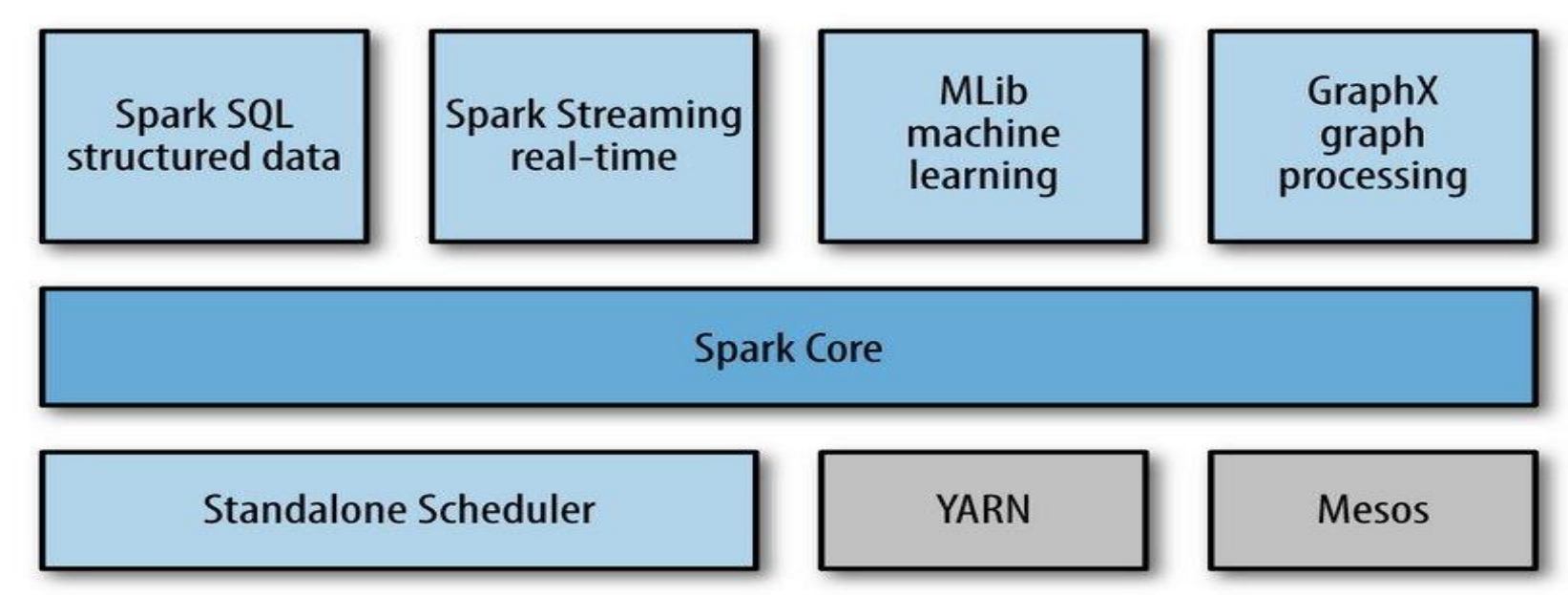
Apache Spark

- Tổng quan về Apache Spark
- Thành phần của Apache Spark
- Tại sao nên sử dụng Apache Spark
- Những tính năng nổi bật
- Quản lý bộ nhớ của Apache Spark
- Ngôn ngữ lập trình trong Spark

Tổng quan về 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
- Cho phép xây dựng các mô hình dự đoán nhanh chóng với việc tính toán được thực hiện trên một nhóm các máy tính
- Spark không chỉ hữu ích cho học máy mà còn cho cả việc xử lý luồng dữ liệu hoàn chỉnh
- Việc tính toán thực hiện:
  - Cùng lúc trên nhiều máy khác nhau, ở bộ nhớ trong (in-memories) hay thực hiện hoàn toàn trên RAM => Tốc độ xử lý của Spark rất nhanh

Thành phần của Apache Spark



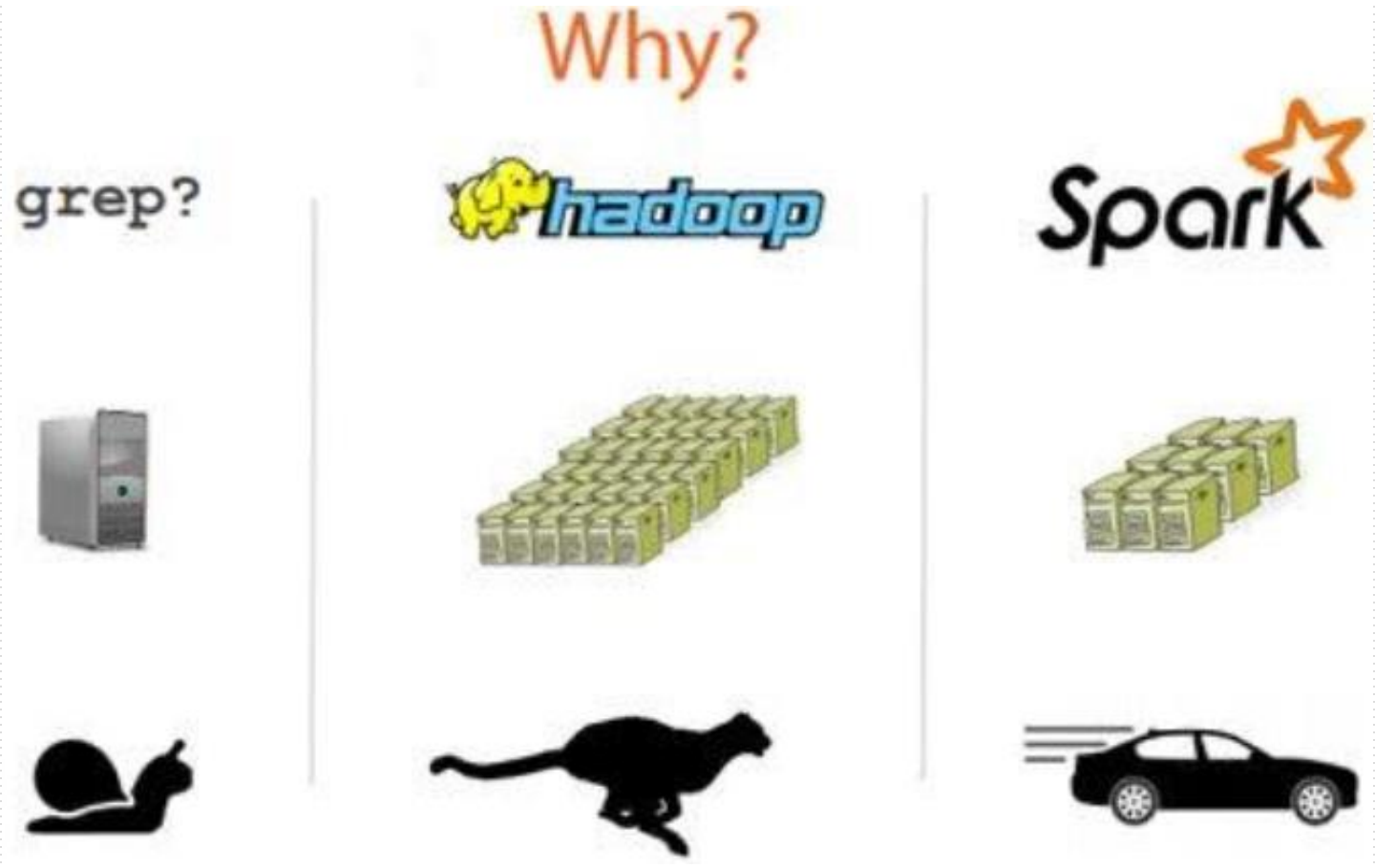
- Thành phần trung của Spark là Spark Core, cung cấp:
  - Những chức năng cơ bản nhất của Spark như lập lịch cho các tác vụ, quản lý bộ nhớ, fault recovery, tương tác với các hệ thống lưu trữ…
  - API để định nghĩa RDD (Resilient Distributed DataSet): là tập hợp của các item được phân tán trên các node của cluster và có thể được xử lý song song

Thành phần của Apache Spark

- Spark có thể chạy trên nhiều loại Cluster Managers như:
  - Hadoop YARN
  - Apache Mesos
  - Standalone Scheduler (được cung cấp bởi Spark)
- Spark SQL:
  - Cho phép truy vấn dữ liệu cấu trúc qua các câu lệnh 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: thư viện để xử lý đồ thị

Tại sao nên sử dụng Apache Spark

- Tốc độ thực thi rất nhanh



Những tính năng nổi bật

- "Spark as a Service": Giao diện REST để quản lí (submit, start, stop, xem trạng thái) spark job, spark context
- Tăng tốc, giảm độ trễ thực thi job xuống mức chỉ tính bằng giây bằng cách tạo sẵn spark context cho các job dùng chung
- Stop job đang chạy bằng cách stop spark context
- Bỏ bước upload gói jar lúc start job làm cho job được start nhanh hơn
- Cung cấp hai cơ chế chạy job đồng bộ và bất đồng bộ
- Cho phép cache RDD theo tên, tăng tính chia sẻ và sử dụng lại RDD giữa các job
- Hỗ trợ viết spark job bằng cú pháp SQL
- Dễ dàng tích hợp với các công cụ báo cáo như: Business Intelligence, Analytics, Data Integration Tools

Quản lý bộ nhớ của Apache Spark

- Spark giải quyết các vấn đề xung quanh định nghĩa Resilient Distributed Datasets (RDDs)
- RDDs hỗ trợ hai kiểu thao tác: transformations và action
  - Thao tác chuyển đổi (tranformation) tạo ra dataset mới từ dataset đã có
  - Thao tác actions trả về giá trị cho chương trình điều khiển (driver program) sau khi thực hiện tính toán trên dataset
- Spark thực hiện đưa các thao tác RDD chuyển đổi vào DAG (Directed Acyclic Graph – Đồ thị định hướng không tuần hoàn) và bắt đầu thực hiện
- Khi một action được gọi trên RDD, Spark sẽ tạo DAG và chuyển cho DAG scheduler
- DAG scheduler chia các thao tác thành các nhóm (stage) khác nhau của các task

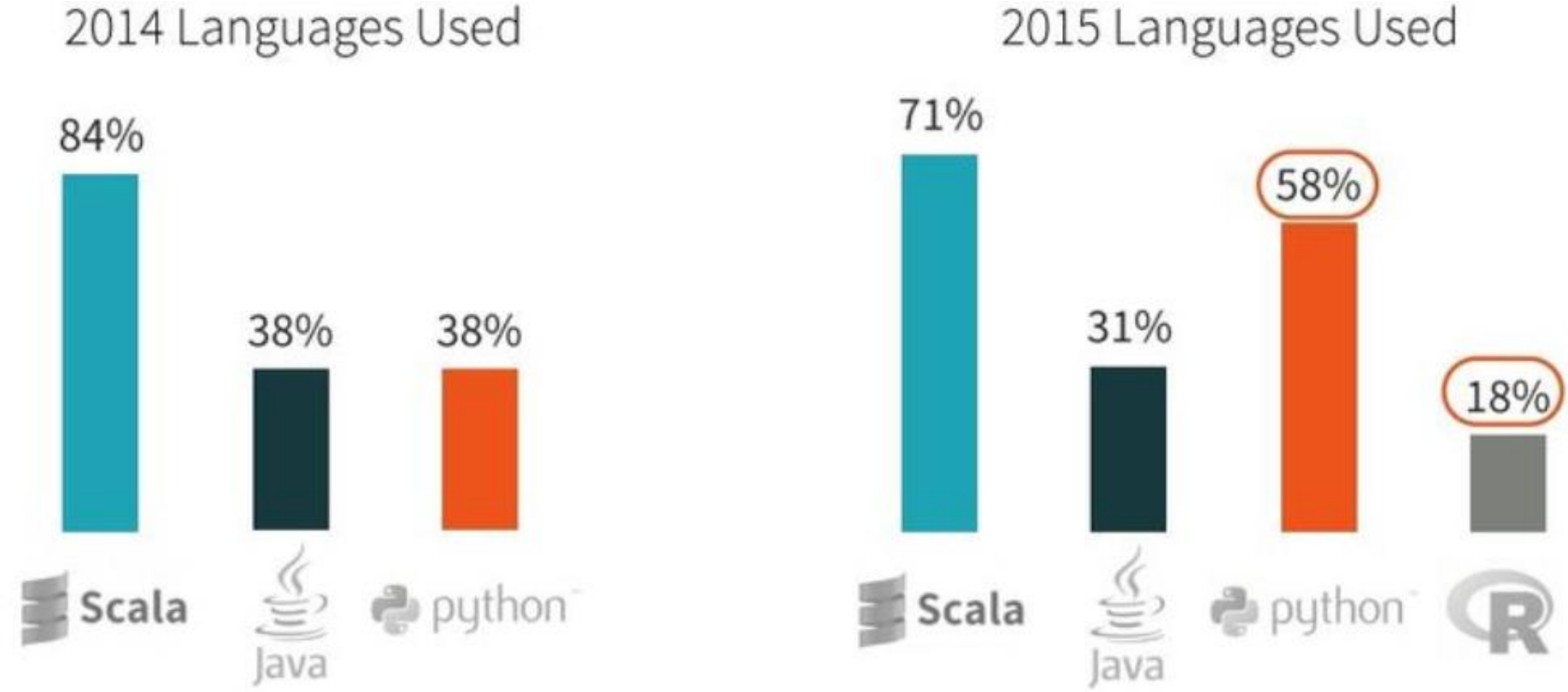
Quản lý bộ nhớ của Apache Spark

- Mỗi Stage bao gồm các task dựa trên phân vùng của dữ liệu đầu vào có thể pipline với nhau và có thể thực hiện một cách độc lập trên một máy worker
- DAG scheduler sắp xếp các thao tác phù hợp với quá trình thực hiện theo thời gian sao cho tối ưu nhất
- Kết quả cuối cùng của DAG scheduler là một tập các stage. Các Stages được chuyển cho Task Scheduler
- Task Scheduler sẽ chạy các task thông qua cluster manager (Spark Standalone/Yarn/Mesos)
- Mỗi Worker bao gồm một hoặc nhiều Excuter
- Các excuter chịu trách nhiệm thực hiện các task trên các luồng riêng biệt

Quản lý bộ nhớ của Apache Spark

- Việc chia nhỏ các task giúp đem lại hiệu năng cao hơn, giảm thiểu ảnh hưởng của dữ liệu không đối xứng (kích thước các file không đồng đều)
- Saprk sử dụng khái niệm là “storage level” để quản lý cấp độ của lưu trữ của dữ liệu:
  - Spark truy cập dữ liệu được lưu trữ ở các nguồn khác nhau như: HDFS, Local Disk, RAM
  - Cache Manager sử dụng Block Manager để quản lý dữ liệu
  - Cache Manager quản lý dữ liệu nào được Cache trên RAM, thường là dữ liệu được sử dụng thường xuyên nhất
  - Nếu kích thước RAM không đủ chứa dữ liệu thì dữ liệu sẽ được lưu trữ sang Tachyon và cuối cùng là lưu trữ lên đĩa
  - Khi dữ liệu (RDD) không được lưu trữ trên RAM, khi có nhu cầu sử dụng đến, chúng sẽ được recompute lại

Ngôn ngữ lập trình trong Spark



Những công ty đang sử dụng Apache Spark

- Hiện nay, có rất nhiều công ty lớn đã dùng Spark như Yahoo, Twitter, Ebay....