

SPARK

1. O que é?

Apache Spark:

- é um framework de código fonte aberto para computação distribuída. (Wikipedia)
- é uma plataforma de computação em *cluster* projetada para ser *rápida* e *de uso geral*. (O'Reilly)

2. História

Foi desenvolvido no AMPLab da Universidade da Califórnia e posteriormente repassado para a Apache Software Foundation que o mantém desde então.

3. Para que serve?

O Spark foi projetado para cobrir uma ampla gama de cargas de trabalho que anteriormente exigiam sistemas distribuídos separados, incluindo aplicativos em lote, algoritmos iterativos, consultas interativas e streaming.

Ao oferecer suporte a essas cargas de trabalho no mesmo mecanismo, o Spark torna fácil e barato *combinar* diferentes tipos de processamento, o que geralmente é necessário em pipelines de análise de dados de produção. Além disso, reduz a carga de gerenciamento de manutenção de ferramentas separadas.

4. Quem utiliza?

Engenheiros de Dados e Cientistas de Dados.

4.1 Engenharia de Dados

Para nossos propósitos aqui, pensamos em engenheiros como uma grande classe de desenvolvedores de software que usam o Spark para criar aplicativos de processamento de dados de produção. Esses desenvolvedores geralmente entendem os princípios da engenharia de software, como encapsulamento, design de interface e programação orientada a objetos. Frequentemente, eles são formados em ciência da computação. Eles usam suas habilidades de engenharia para projetar e construir sistemas de software que implementam um caso de uso de negócios.

Para engenheiros, o Spark fornece uma maneira simples de paralelizar esses aplicativos em clusters e oculta a complexidade da programação de sistemas distribuídos, comunicação de rede e tolerância a falhas. O sistema oferece a eles controle suficiente para monitorar, inspecionar e ajustar aplicativos, permitindo que implementem tarefas comuns rapidamente. A natureza modular da API (baseada na passagem de coleções distribuídas de objetos) torna fácil fatorar o trabalho em bibliotecas reutilizáveis e testá-lo localmente.

Os usuários do Spark optam por usá-lo para seus aplicativos de processamento de dados porque ele oferece uma ampla variedade de funcionalidades, é fácil de aprender e usar, além de ser maduro e confiável.

4.2 Data Science

Embora não haja uma definição padrão, para nossos propósitos, um *cientista de dados* é alguém cuja principal tarefa é analisar e modelar dados. Os cientistas de dados podem ter experiência com SQL, estatística, modelagem preditiva (aprendizado de máquina) e programação, geralmente em Python, Matlab ou R.

Os cientistas de dados usam suas habilidades para analisar dados com o objetivo de responder a uma pergunta ou descobrir insights. Muitas vezes, seu fluxo de trabalho envolve análise *ad hoc*, então eles usam shells interativos (em vez de criar aplicativos complexos) que permitem ver os resultados de consultas e trechos de código no menor tempo possível. A velocidade e as APIs simples do Spark brilham para essa finalidade, e suas bibliotecas integradas significam que muitos algoritmos estão disponíveis imediatamente.

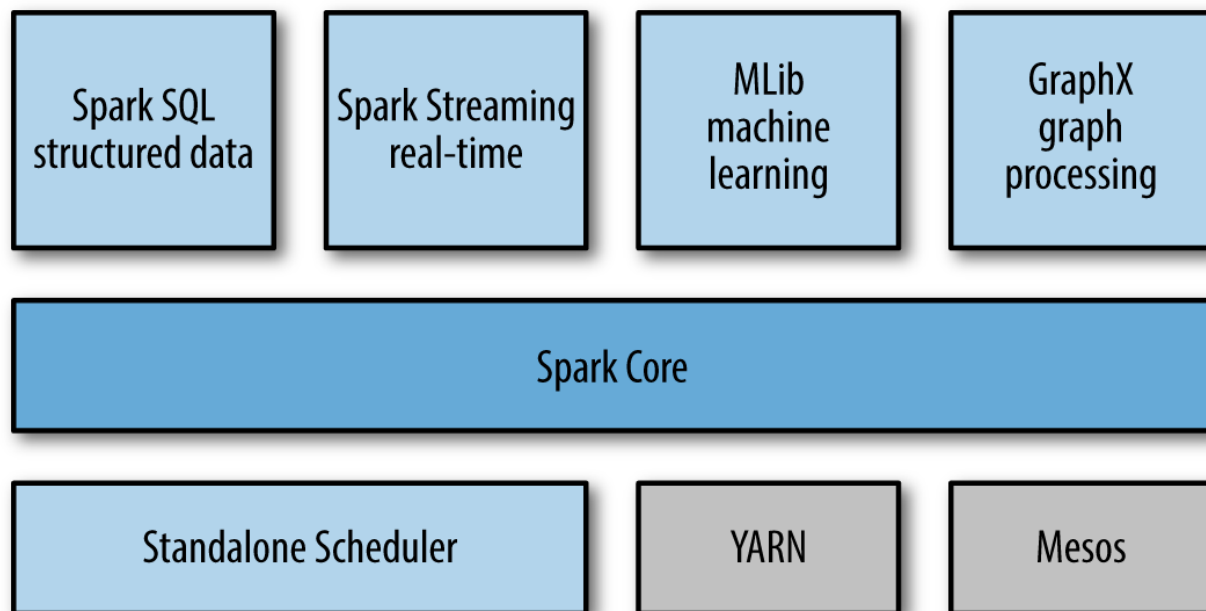
O Spark suporta as diferentes tarefas da ciência de dados com vários componentes. Com o shell Spark é fácil fazer análises interativas de dados usando Python ou Scala. O Spark SQL também possui um shell SQL que pode ser usado para fazer exploração de dados, ou o Spark SQL pode ser usado como parte de um programa regular do Spark ou no shell do Spark. O aprendizado de máquina e a análise de dados são suportados por meio das bibliotecas MLlib. Além disso, há suporte para chamar programas externos em Matlab ou R. O Spark permite que os cientistas de dados resolvam problemas com tamanhos de dados maiores do que antes com ferramentas como R ou Pandas.

5. Arquitetura

A arquitetura do Spark é definida por uma pilha unificada de componentes integrados.

Em sua essência, o Spark é um “mecanismo computacional” responsável por agendar, distribuir e monitorar aplicativos que consistem em diversas tarefas computacionais sendo executadas em inúmeras máquinas de trabalho (que juntas compõem um *cluster computacional*).

5.1 Componentes



- **Spark Core**

Spark Core contém a funcionalidade básica do Spark, incluindo componentes para agendamento de tarefas, gerenciamento de memória, recuperação de falhas, interação com sistemas de armazenamento e muito mais.

- **Spark SQL**

Spark SQL é o pacote do Spark para trabalhar com dados estruturados. Permite a consulta de dados via SQL assim como a Variante Apache Hive do SQL — chamada Hive Query Language (HQL) — e suporta muitas fontes de dados, incluindo tabelas Hive, Parquet e JSON.

- **Spark Streaming**

O Spark Streaming é um componente do Spark que permite o processamento de fluxos de dados ao vivo.

- **MLlib**

Spark vem com uma biblioteca contendo funcionalidades de aprendizado de máquina (ML) comuns, chamado MLlib. O MLlib fornece vários tipos de algoritmos de aprendizado de máquina, incluindo classificação, regressão, agrupamento e filtragem colaborativa, além de oferecer suporte a funcionalidades como avaliação de modelo e importação de dados.

- **GraphX**

GraphX é uma biblioteca para manipular gráficos (por exemplo, o gráfico de amigos de uma rede social) e realizar cálculos paralelos de gráficos.

- **Gerenciadores de Clusters**

O Spark foi projetado para escalar com eficiência de um para muitos milhares de nós de computação. Para conseguir isso e maximizar a flexibilidade, o Spark pode executar uma variedade de *gerenciadores de cluster*, incluindo Hadoop YARN, Apache Mesos e um gerenciador de *cluster* simples incluído no próprio Spark chamado Standalone Scheduler

6. Linguagens de Programação

Possui APIs de alto nível para linguagens Scala, Python, Java, R.

Possui módulo SQL.

Fonte:

Spark Overview

Apache Spark is a unified analytics engine for large-scale data processing. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs.

★ <https://spark.apache.org/docs/latest/index.html>



Learning Spark

This chapter provides a high-level overview of what Apache Spark is. If you are already familiar with Apache Spark and its components, feel free to jump ahead to Chapter 2. Apache Spark is a cluster computing

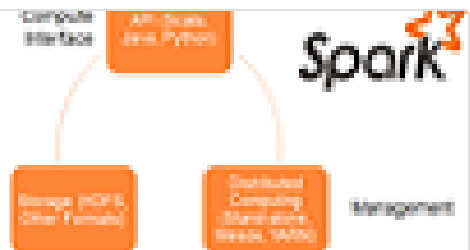
🔴 <https://www.oreilly.com/library/view/learning-spark/9781449359034/ch01.html>



Big Data com Apache Spark - Parte 1: Introdução

O Apache Spark é um framework de big data construído para ser veloz, fácil de usar e com análises sofisticadas. Nesse artigo, Srin Penchikala mostra como o Spark ajuda no processamento e análise

InfoQ <https://www.infoq.com/br/articles/apache-spark-introduction/>



An Introduction to Apache, PySpark and Dataframe Transformations


Apache arises as a new engine and programming model for data analytics. It's origin goes back to 2009, and the main reasons why it has gained so much importance in the past recent years are due to changes in economic

tds <https://towardsdatascience.com/an-introduction-to-apache-pyspark-and-dataframe-transformations-2a6d4229f0e3>



Big Data Analysis: Spark and Hadoop

According to Forbes, about 2.5 quintillion bytes of data is generated every day. Nonetheless, this number is just projected to constantly increase in the following years (90% of nowadays stored data has

 <https://towardsdatascience.com/big-data-analysis-spark-and-hadoop-p-a11ba591c057>

