Curso 03: Administração de Dados Complexos em Larga Escala

-- Comparação Hadoop x Spark --

Prof. Jose Fernando Rodrigues Junior

Objetivo: comparar as soluções de processamento Big Data Hadoop e Spark

Níveis da Análise de Dados

- **Análise de dados básica**: contagens, somas, médias, máximo, mínimo, e ordenação;
- **Análise de dados estatística**: distribuição de dados, ajuste de modelo, teste de hipóteses, métricas, etc;
- **Análise de dados avançada**: aprendizado de máquina, classificação, regressão, recomendação, clusterização, etc;
- **Aprendizado de máquina avançado**: arquiteturas de redes neurais visando inteligência artificial.

Níveis da Análise de Dados

- **Análise de dados básica**: [illegible]s, médias, máximo, mínimo, e ord[illegible]
- **Análise de dados estatística**: distribuição de dados, ajuste de modelo, t[illegible]s, métricas, etc;
- **Análise de dados avançada**: aprendizado de máquina, classificação, regres[illegible]o, clusterização, etc;
- **Aprendizado de máquina avançado**: arquiteturas de redes neurais visando inteligê[illegible]

Curso 03/11 - DW/OLAP

Curso 02/05

Curso 02/03 - Spark

Curso 05/07/08/10

Comparativo

vs

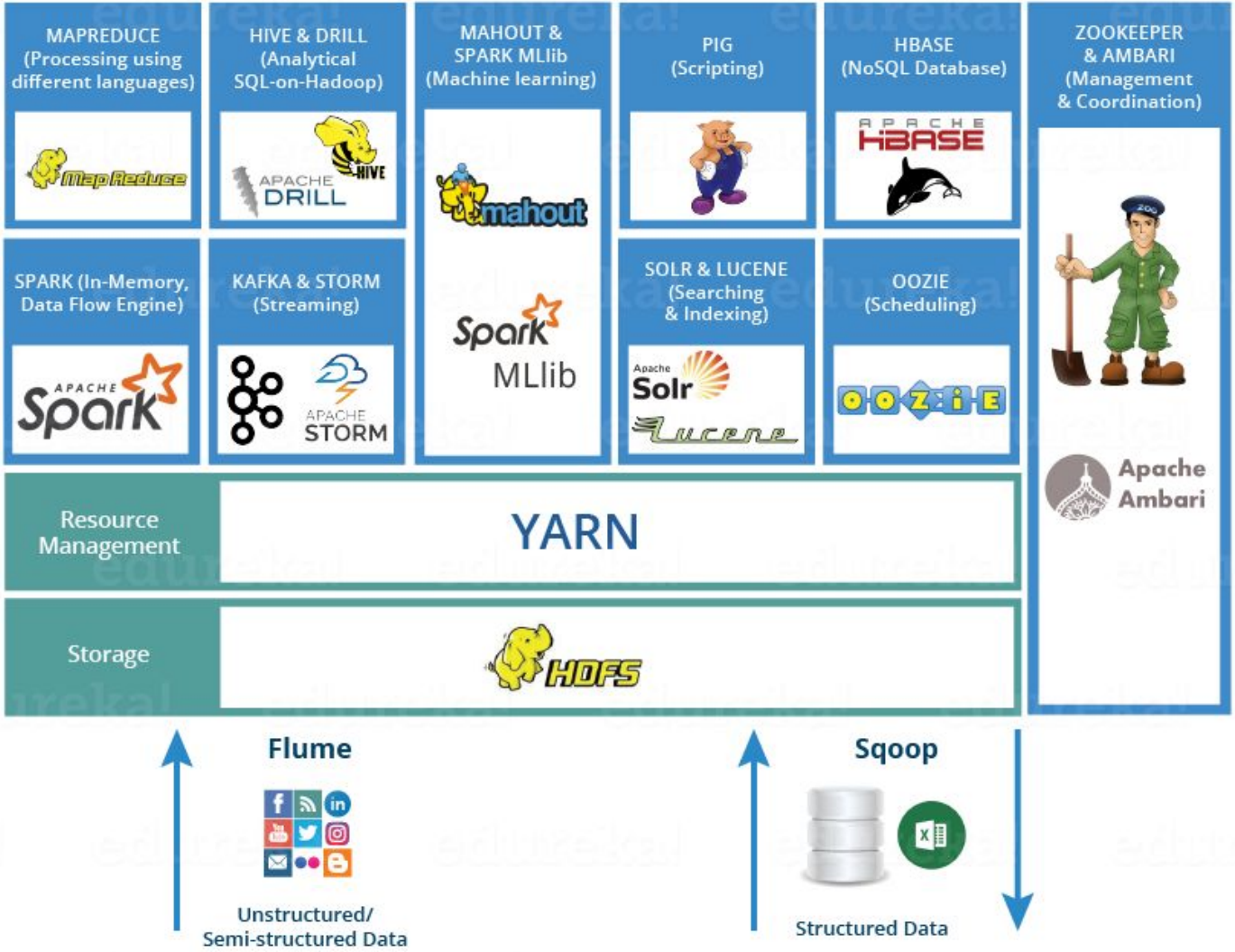
Hadoop

- O Apache Hadoop é um **arcabouço de código aberto** para armazenar e processar com eficiência conjuntos de dados que variam de gigabytes a petabytes;
- um projeto que ganhou força já a partir de 2005;
- Sua escala é conhecida por abranger **Petabytes de dados** - Facebook, Yahoo, Expedia, British Airways,….;
- Dezenas de outros softwares **trabalham em conjunto** ou sobre a combinação HDFS+MapReduce: Hive, HBase, Kylin, Mahout, Pig, etc

⇒ The Hadoop Ecosystem Table

⇒ Hadoop at Amazon AWS

É tudo sobre Big Data.



Hadoop

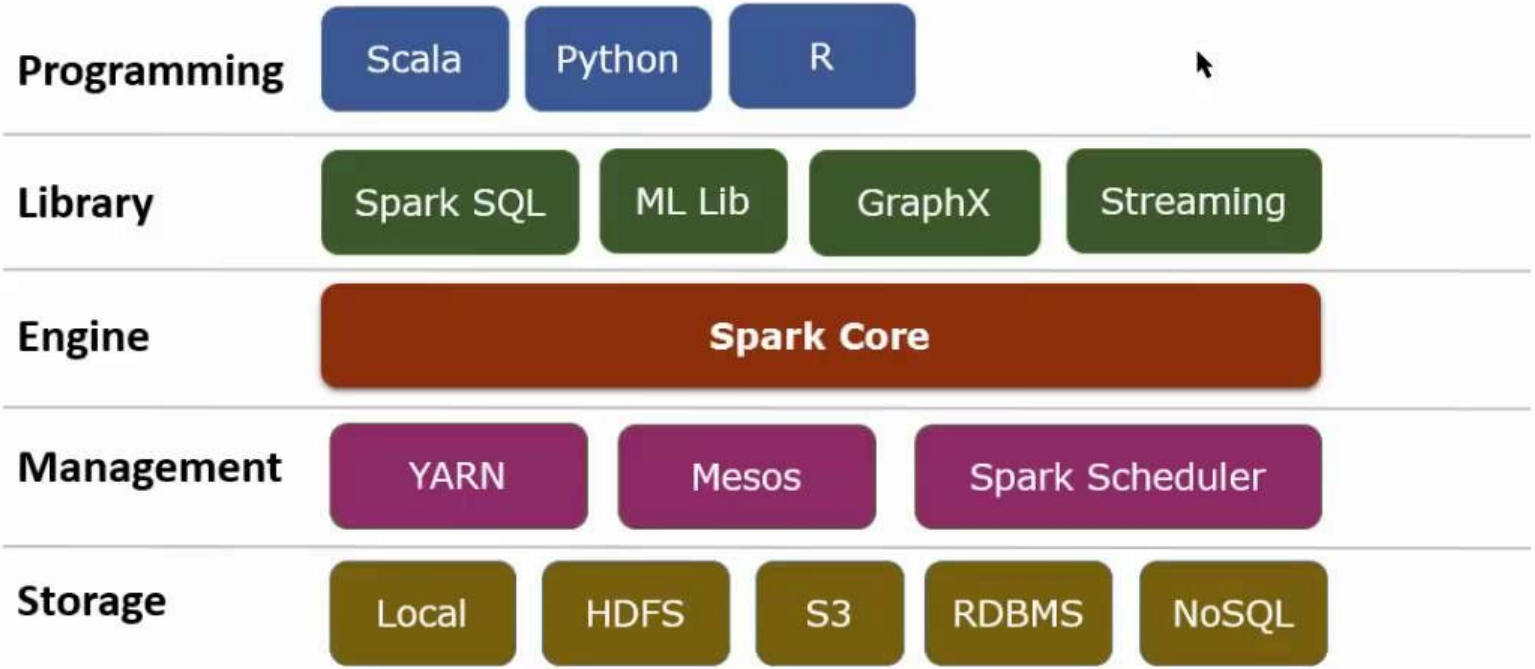
⇒ Limitações:

- não é a melhor solução para **poucos dados** - alguns Terabytes por exemplo;
- não funciona de modo **interativo**, mas em processamento batch - inadequado para *real-time processing*;
- difícil integração com dados produzidos de maneira contínua (***streaming***);
- processamento **iterativo** dificultado, ou inviável;
- uso mais **complexo** do que Spark - Java apenas, e projeto complexo.

Spark

- Arcabouço de **processamento distribuído**;
- Faz uso de processamento em memória, ao invés de fluxo sequencial de dados - 10 a 100 vezes **mais rápido do que Hadoop**;
- Consegue **sanar as limitações** do Hadoop:
  - linguagens de programação: Python, Scala, Java, e R;
  - lida bem com dados em streaming;
  - faz processamento iterativo ou batch.
- **Integrável** com diversos subsistemas: HDFS, HBase, Yarn, Mesos, Cassandra, ou standalone.

Spark Framework



Comparativo

Hadoop	Spark
Processamento +Armazenamento distribuídos	Processamento distribuído
MapReduce	Resilient Distributed Dataset (RDD)
Disk-intensive	Memory-intensive
Batch	Iterativo (grafos, proces. numérico) e Interativo (real-time, spark shell); e também batch
Java (embora existam outras iniciativas)	Scala, Python, Java, e R (simplicidade de uso)

Comparativo

Task	Hadoop	Spark
Data Warehouse	Hive	Spark SQL
ETL	Sqoop, Flume	Spark SQL
Aprendizado de máquina	Mahout	Machine Learning Library (MLlib)
Streaming	Storm e/ou Kafka	Spark Streaming
Real-time analytics	None	Suporta
Grafos	Inviável	GraphX

Spark sobre o Hadoop

- Embora versátil, o Spark surgiu no contexto do **ecossistema Hadoop** - uma evolução;
- Aceita outras integrações, mas, comumente, faz uso do **HDFS e do YARN**;
- Hadoop = HDFS+MapReduce, o que torna o conceito "usa Hadoop" **nebuloso**;
- O Spark não é um substituto para o Hadoop por duas razões:
  - usa parte de seu legado;
  - ainda que uma evolução, sua escala é menor.

Spark

⇒ Limitações

- requer que **memória disponível** nos nós de processamento - se tiver que fazer paginação (*virtual memory*), o desempenho cai drasticamente;

- a necessidade de **memória eleva o custo**; por exemplo, se deseja-se um cluster com máquinas de baixo custo, o Hadoop é uma solução mais adequada.

Conclusões

- **Hadoop**, muito bom em:
  - processamento simples;
  - altíssima escala;
  - sobre clusters de baixo custo;
- **Spark**, muito bom em:
  - processamento simples ou complexo;
  - alta escala;
  - sobre clusters de custo mais elevado.

⇒ **Critérios**:

- qual a escala dos meus dados?
- quais computadores eu já tenho?
- quais problemas eu preciso resolver?

Conclusões

algumas linhas | KBytes | MBytes | GBytes | Terabytes | Centenas de TBytes | Petabytes | Centenas de Petabytes

Conclusões

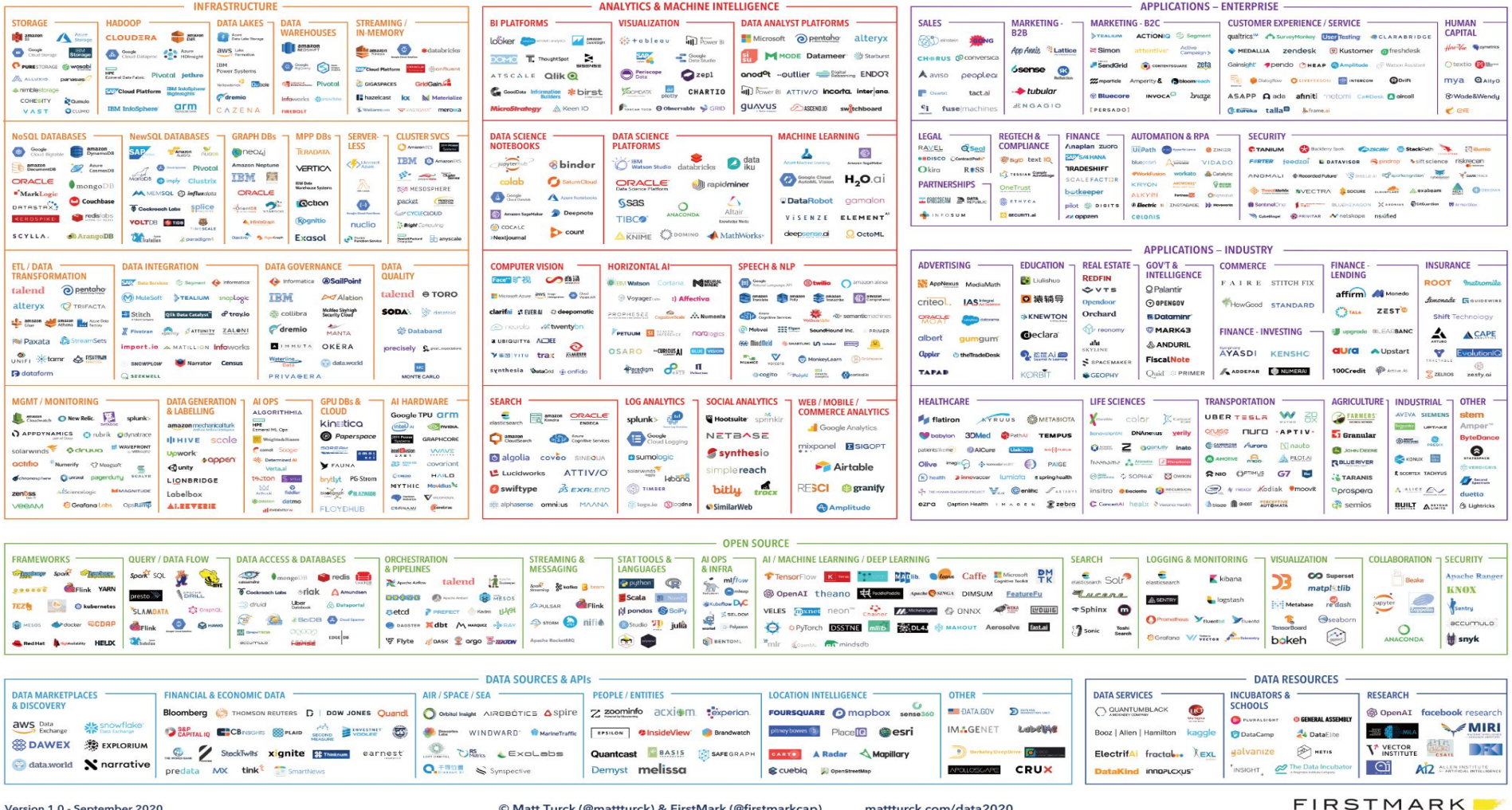
algumas linhas

KBy

s de

ces

Data & IA Landscape 2020



mattturck.com/data2020

FIRSTMARK
EARLY STAGE VENTURE CAPITAL