Senac

Fecomércio

Sesc

Big Data

Prof. Marco Mialaret

Maio

2024

Big Data

Onde me encontrar:

https://www.linkedin.com/in/marco-mialaret-junior/

e

https://github.com/MatmJr

Apache Spark

Big Data

O Apache Spark é um sistema de processamento distribuído de código aberto para big data. Utiliza armazenamento em cache na memória e execução otimizada para consultas analíticas rápidas em dados de qualquer tamanho.

Big Data

Oferece APIs em Java, Scala, Python e R, permitindo a reutilização de código em várias workloads, como processamento de lotes, consultas interativas, análises em tempo real, machine learning e processamento de gráficos.

Big Data

Origem e Início:

- Iniciado em 2009 como um projeto de pesquisa no AMPLab da UC Berkeley.
- Colaboração entre estudantes, pesquisadores e professores.

Big Data

Objetivo:

- Criar uma estrutura otimizada para processamento iterativo rápido, como machine learning e análise interativa de dados.
- Escalabilidade e a tolerância a falhas.

Big Data

Primeiro Artigo:

- "Spark: Cluster Computing with Working Sets" publicado em junho de 2010.
- Spark lançado como código aberto sob licença BSD.

Big Data

Transição para Apache:

- Tornou-se um projeto de incubação na Apache Software Foundation (ASF) em junho de 2013.
- Estabelecido como um projeto de alto nível da Apache em fevereiro de 2014.

Funcionamento do Spark

Big Data

Antes de 2009, a principal ferramenta de processamento de dados era o Hadoop MapReduce, um modelo de programação projetado para big data que utiliza algoritmos distribuídos e paralelos. Ele permite que desenvolvedores criem operadores altamente paralelizados sem se preocuparem com a distribuição do trabalho e a tolerância a falhas.

Big Data

No entanto, um desafio era o processo sequencial de várias etapas necessário para executar um trabalho. Pois, em cada etapa, o MapReduce lê dados do cluster, executa operações e grava os resultados no HDFS. Isso resulta em lentidão devido à latência da E/S do disco, pois cada etapa exige leitura e gravação no disco.

Big Data

O Spark foi criado para resolver as limitações do MapReduce, processando dados na memória e reduzindo etapas de tarefas. Ele executa operações em uma única etapa, o que acelera a execução.

Big Data

Utiliza cache na memória para reutilizar dados em várias operações, especialmente em algoritmos de machine learning, através de DataFrames e RDDs (Resilient Distributed Datasets). Isso reduz drasticamente a latência, tornando o Spark várias vezes mais rápido que o MapReduce.

Big Data

Embora o design do Spark e do Hadoop MapReduce seja diferente, muitas organizações os utilizam juntos para resolver desafios comerciais mais amplos.

O Spark é uma estrutura de código aberto voltada para consultas interativas, machine learning e workloads em tempo real. Ele executa análises em sistemas de armazenamento como HDFS, Amazon Redshift, Amazon S3, Couchbase e Cassandra.

Big Data

No Hadoop, o Spark usa o YARN para compartilhar clusters e conjuntos de dados comuns com outros mecanismos do Hadoop, garantindo serviços e respostas consistentes.

Benefícios do Spark

Big Data

Rápido

Por meio do armazenamento em cache na memória e execução otimizada de consultas, o Spark pode oferecer consultas analíticas rápidas de dados de qualquer tamanho.

Big Data

Para desenvolvedores

O Apache Spark sustenta de modo nativo Java, Scala, R e Python, oferecendo a você várias linguagens para a criação de aplicativos. Essas APIs facilitam as coisas para seus desenvolvedores, pois ocultam a complexidade do processamento distribuído por trás de operadores simples e de alto nível que reduzem drasticamente a quantidade de código necessária.

Big Data

Várias workloads

O Apache Spark vem com a capacidade de executar várias workloads, incluindo consultas interativas, análises em tempo real, machine learning e processamento de gráficos. Uma aplicação pode combinar várias workloads facilmente.

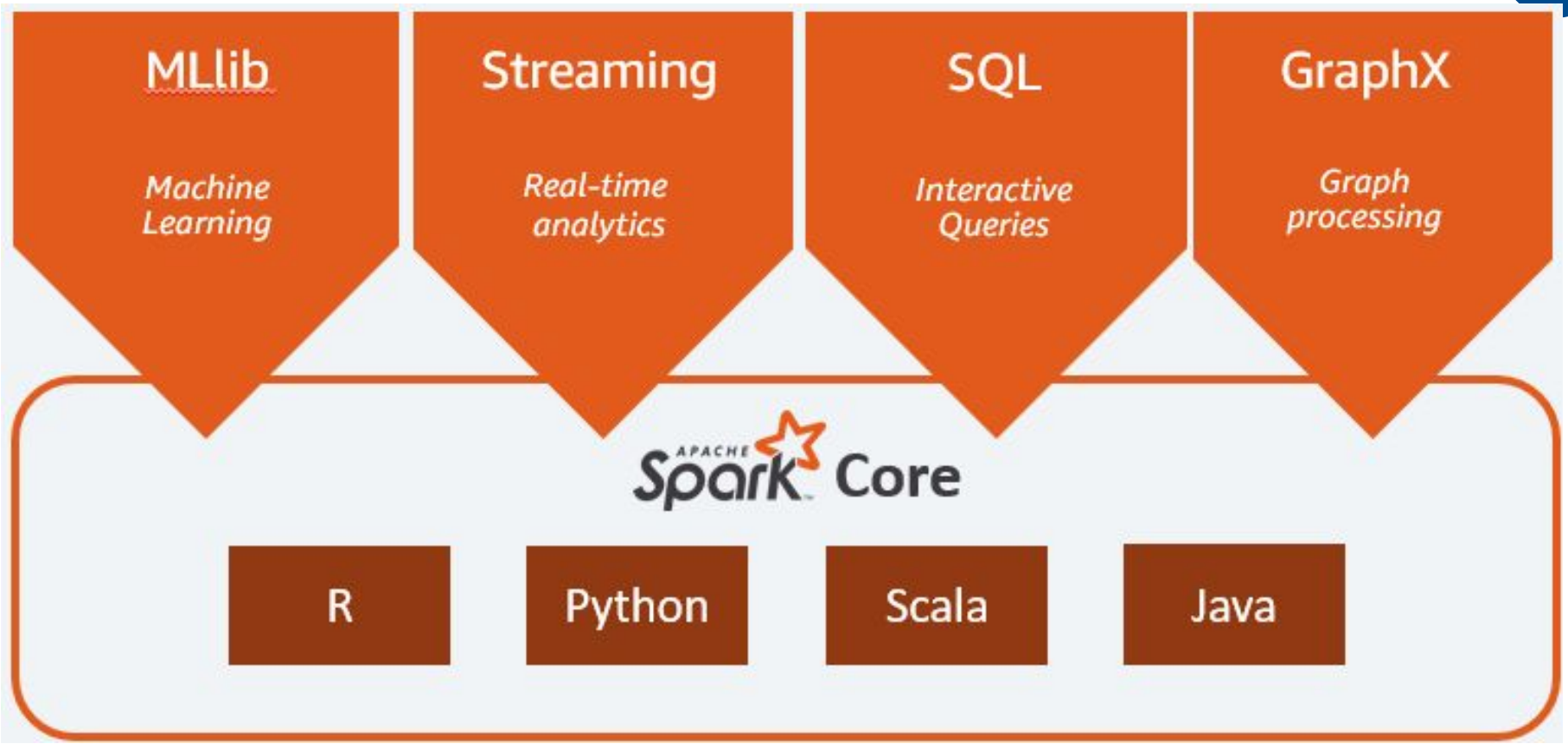
Workloads do Apache Spark

Big Data

A estrutura do Spark inclui:

- **Spark Core** como base para a plataforma
- **Spark SQL** para consultas interativas
- **Spark Streaming** para análises em tempo real
- **Spark MLlib** para machine learning
- **Spark GraphX** para processamento de gráficos

Big Data


MLlib
Machine Learning
Streaming
Real-time analytics
SQL
Interactive Queries
GraphX
Graph processing
APACHE Spark Core
R
Python
Scala
Java

Big Data

Spark Core

O Spark Core é a base da plataforma, gerenciando memória, recuperação de falhas, programação, distribuição e monitoramento de tarefas, além de interagir com sistemas de armazenamento. Ele oferece APIs para Java, Scala, Python e R, simplificando o processamento distribuído com operadores de alto nível.

Big Data

MLlib

O Spark inclui o MLlib, uma biblioteca de algoritmos de machine learning para dados em grande escala. Cientistas de dados podem treinar modelos com R ou Python, salvar com MLlib e importar para pipelines em Java ou Scala.

Big Data

Projetado para computação rápida e interativa na memória, o Spark permite execução rápida de machine learning. Os algoritmos disponíveis incluem classificação, regressão, agrupamento, filtragem colaborativa e mineração de padrões.

Big Data

Spark Streaming

Solução em tempo real que utiliza o agendamento rápido do Spark Core para análises de streaming. Ele processa dados em minilotes, permitindo o uso do mesmo código de aplicação para análises em lotes e streaming, aumentando a produtividade do desenvolvedor.

Big Data

O Spark Streaming é compatível com dados de Twitter, Kafka, Flume, HDFS, ZeroMQ, entre outros do ecossistema Spark.

Big Data

Spark SQL

É um mecanismo de consulta distribuído que oferece consultas interativas de baixa latência, até 100 vezes mais rápidas que o MapReduce. Inclui um otimizador baseado em custos, armazenamento colunar e geração de código para consultas rápidas e escaláveis em milhares de nós.

Big Data

Permite que analistas usem SQL padrão ou Hive Query Language e desenvolvedores utilizem APIs em Scala, Java, Python e R. Suporta várias fontes de dados, como JDBC, ODBC, JSON, HDFS, Hive, ORC, Parquet, Amazon Redshift, Amazon S3, Couchbase, Cassandra, MongoDB, Salesforce.com e Elasticsearch.

Big Data

Spark GraphX

O Spark GraphX é uma estrutura distribuída de processamento de gráficos construída sobre o Spark. Ela oferece ETL, análise exploratória e computação gráfica iterativa, permitindo a criação e transformação interativa de grandes estruturas de dados gráficos. O GraphX inclui uma API flexível e uma seleção de algoritmos gráficos distribuídos.

Big Data

Mais detalhes:

https://aws.amazon.com/pt/what-is/apache-spark/

O PySpark

Big Data

PySpark é uma API em Python para executar o Spark e foi lançado para oferecer suporte à colaboração entre Apache Spark e Python. O PySpark também oferece suporte à interface do Apache Spark com conjuntos de dados distribuídos resilientes (RDDs) na linguagem de programação Python.

Big Data

O PySpark também pode ser utilizado no Colab. Vamos começar a explorar esse novo mundo juntos:

Crie um notebook novo

Big Data

Use a primeira célula para instalar a biblioteca do PySpark

```
!pip install pyspark
```

Big Data

Vamos iniciar uma sessão Spark

```
from pyspark.sql import SparkSession

spark =
SparkSession.builder.appName("Spark-ETL").getOrCreate()
```

Big Data

Carregando o primeiro dataset

```
import pyspark.sql.functions as F

data = spark.read.csv("sample_data/california_housing_train.csv",
header=True)

print(f"The data contains: {data.count()} rows")
data.show()
```

Big Data

The data contains: 17000 rows

longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
-114.310000	34.190000	15.000000	5612.000000	1283.000000	1015.000000	472.000000	1.493600	66900.000000
-114.470000	34.400000	19.000000	7650.000000	1901.000000	1129.000000	463.000000	1.820000	80100.000000
-114.560000	33.690000	17.000000	720.000000	174.000000	333.000000	117.000000	1.650900	85700.000000
-114.570000	33.640000	14.000000	1501.000000	337.000000	515.000000	226.000000	3.191700	73400.000000
-114.570000	33.570000	20.000000	1454.000000	326.000000	624.000000	262.000000	1.925000	65500.000000
-114.580000	33.630000	29.000000	1387.000000	236.000000	671.000000	239.000000	3.343800	74000.000000
-114.580000	33.610000	25.000000	2907.000000	680.000000	1841.000000	633.000000	2.676800	82400.000000
-114.590000	34.830000	41.000000	812.000000	168.000000	375.000000	158.000000	1.708300	48500.000000
-114.590000	33.610000	34.000000	4789.000000	1175.000000	3134.000000	1056.000000	2.178200	58400.000000
-114.600000	34.830000	46.000000	1497.000000	309.000000	787.000000	271.000000	2.190800	48100.000000
-114.600000	33.620000	16.000000	3741.000000	801.000000	2434.000000	824.000000	2.679700	86500.000000
-114.600000	33.600000	21.000000	1988.000000	483.000000	1182.000000	437.000000	1.625000	62000.000000
-114.610000	34.840000	48.000000	1291.000000	248.000000	580.000000	211.000000	2.157100	48600.000000
-114.610000	34.830000	31.000000	2478.000000	464.000000	1346.000000	479.000000	3.212000	70400.000000
-114.630000	32.760000	15.000000	1448.000000	378.000000	949.000000	300.000000	0.858500	45000.000000
-114.650000	34.890000	17.000000	2556.000000	587.000000	1005.000000	401.000000	1.699100	69100.000000
-114.650000	33.600000	28.000000	1678.000000	322.000000	666.000000	256.000000	2.965300	94900.000000
-114.650000	32.790000	21.000000	44.000000	33.000000	64.000000	27.000000	0.857100	25000.000000
-114.660000	32.740000	17.000000	1388.000000	386.000000	775.000000	320.000000	1.204900	44000.000000
-114.670000	33.920000	17.000000	97.000000	24.000000	29.000000	15.000000	1.265600	27500.000000

only showing top 20 rows

Dúvidas?

Marco Mialaret, MSc

Telefone:
81 98160 7018

E-mail:
marcomialaret@gmail.com