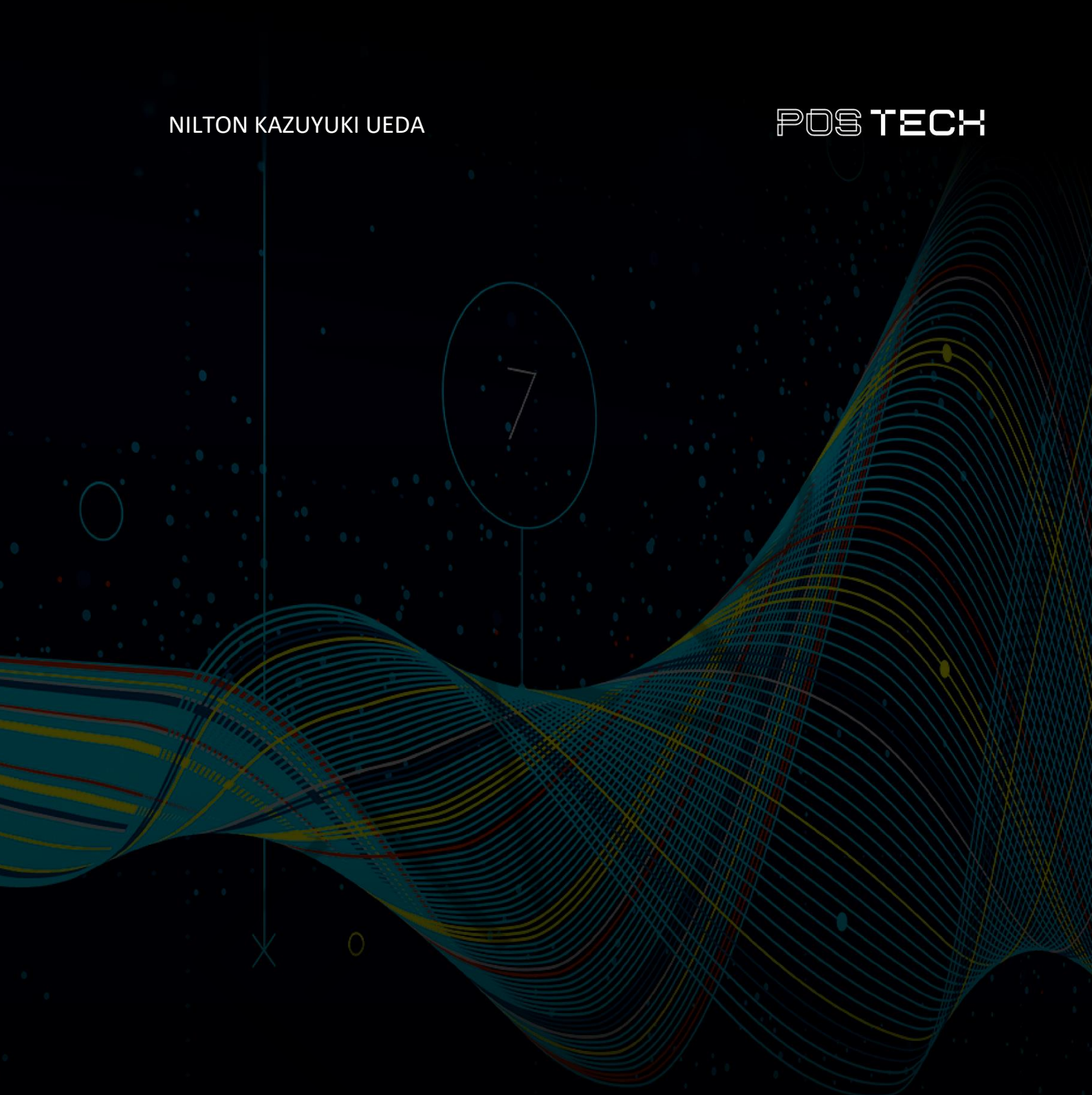
NILTON KAZUYUKI UEDA



DATA ANALYTICS

FRAMEWORK DE BIG DATA

AULA 03

SUMÁRIO

O QUE VEM POR AÍ?

No mundo dos dados (que está sempre em constante evolução), a capacidade de extrair informações relevantes é essencial para tomar decisões informadas e impulsionar o crescimento dos negócios. Nesse contexto, o Apache Spark se destaca como uma poderosa ferramenta de processamento distribuído que permite a realização de consultas e seleções avançadas em grandes conjuntos de dados. Se você se interessa em aprender sobre consultas e seleções no Spark, aproveite esta aula introdutória, que desvendará os segredos dessa técnica.

Você será guiado(a) pelos conceitos fundamentais do Apache Spark e aprenderá a como realizar consultas e seleções de dados em tempo real. Desde a criação de tabelas temporárias e visualizações, até a aplicação de cláusulas de filtragem, ordenação e agregação. Você descobrirá como o Spark pode ser utilizado para obter insights valiosos a partir de conjuntos de dados complexos.

Ao longo da disciplina, você terá a oportunidade de praticar consultas e seleções utilizando a linguagem de programação SQL ou DSL (Domain-Specific Language) do Spark. Aprenderá a escrever consultas eficientes e a utilizar as poderosas funções do Apache para processar e manipular dados em escala.

Prepare-se para explorar o mundo das consultas e seleções, e desbloquear o potencial do processamento distribuído para análise de dados. Com essa aula introdutória, você estará preparado(a) para enfrentar desafios analíticos complexos e tomar decisões baseadas em informações valiosas, em tempo real. Com esse conhecimento, você estará pronto(a) para extrair insights significativos e impulsionar o sucesso dos seus projetos de análise de big data.

HANDS ON

Vamos conhecer mais sobre os conceitos que veremos na aula e colocar a mão na massa?! Assista a videoaula e aprenda conosco!

SAIBA MAIS

O QUE É O SPARK SQL

O Spark SQL é um módulo do Apache Spark, um poderoso framework de processamento distribuído e análise de dados em larga escala. Ele permite que as pessoas desenvolvedoras trabalhem com dados estruturados usando SQL (Structured Query Language) e com datasets distribuídos, combinando os benefícios do processamento em lote e em tempo real.

Fornece uma API unificada para trabalhar com dados estruturados, permitindo que os(as) desenvolvedores(as) usem consultas SQL tradicionais, juntamente com recursos avançados de análise de dados. Ele suporta uma ampla variedade de fontes de dados, incluindo arquivos CSV, JSON, Parquet, bancos de dados relacionais e muito mais. Além disso, o Spark SQL pode ser integrado a outros componentes do ecossistema Spark, como o Spark Streaming, MLlib e GraphX.

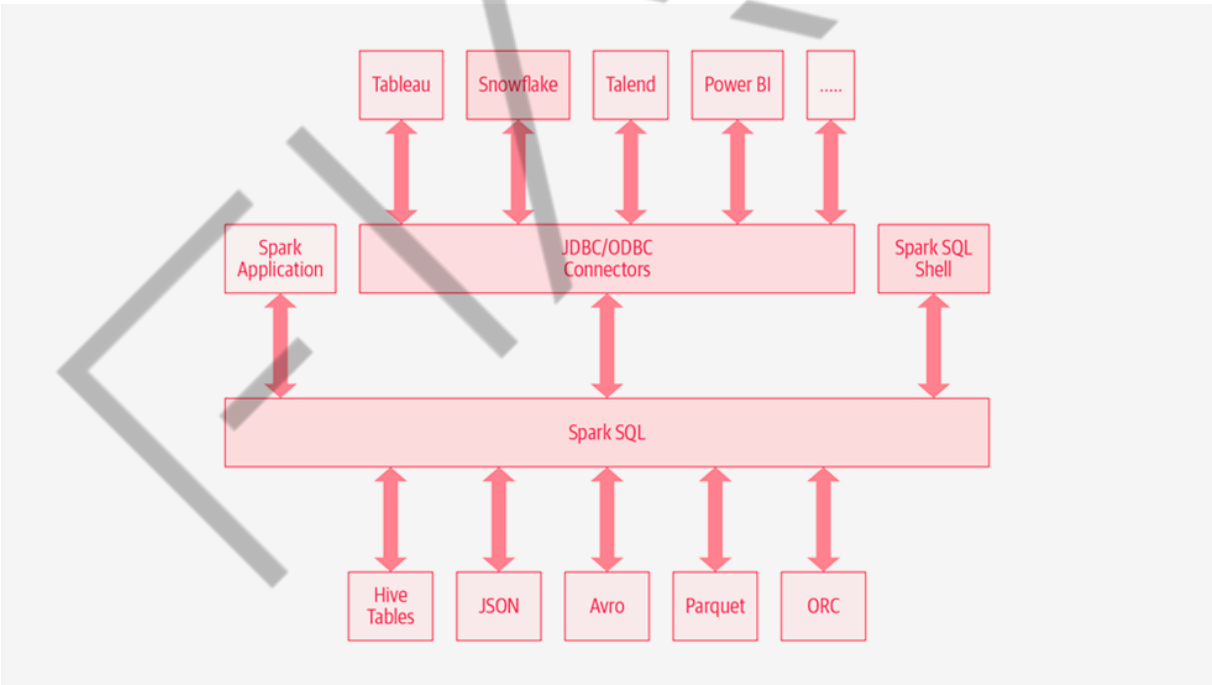


Figura 1 – SparkSQL
Fonte: Learning Spark [s.d.], adaptada por FIAP (2023)

Uma das principais vantagens é sua capacidade de realizar otimizações de consulta para melhorar o desempenho com um otimizador de consultas que pode analisar consultas SQL, otimizá-las e gerar um plano de execução eficiente. Ele também pode aproveitar a execução distribuída para processar consultas em paralelo

em um cluster de computadores, o que permite lidar com grandes volumes de dados de maneira escalável e eficiente.

Outro recurso importante é a capacidade de executar operações de processamento de dados complexas, como agregações, junções, filtros e transformações, tanto em dados estruturados quanto em dados semiestruturados. Ele também suporta funções de agregação personalizadas, permitindo que os desenvolvedores definam suas próprias operações de resumo de dados.

Além de trabalhar com dados estruturados, o Spark SQL oferece suporte a DataFrames e Datasets, que são abstrações de alto nível para manipulação de dados, de acordo com a linguagem de programação escolhida. Os DataFrames são estruturas de dados imutáveis semelhantes a tabelas em um banco de dados relacional, enquanto os Datasets são uma extensão dos DataFrames, que fornecem uma interface tipada para manipulação de dados usando a linguagem Scala, Java ou Python.

O Spark SQL é amplamente utilizado em vários cenários, como análise de dados, processamento de logs, business intelligence e data warehousing. Sua combinação de recursos SQL, processamento distribuído e otimizações de consultas torna-o uma escolha popular para lidar com grandes volumes de dados em ambientes distribuídos.

Pode-se concluir fundamentalmente que o Spark SQL é um componente essencial do ecossistema Spark, fornecendo recursos avançados para consulta, análise e processamento distribuído de dados estruturados. Sua API unificada e capacidades de otimização de consultas o tornam uma ferramenta poderosa para lidar com análise de dados em grande escala.

COMPONENTES DO SPARK SQL

Os dois principais componentes ao se utilizar o Spark SQL são o DataFrame e o SQLContext contidos.

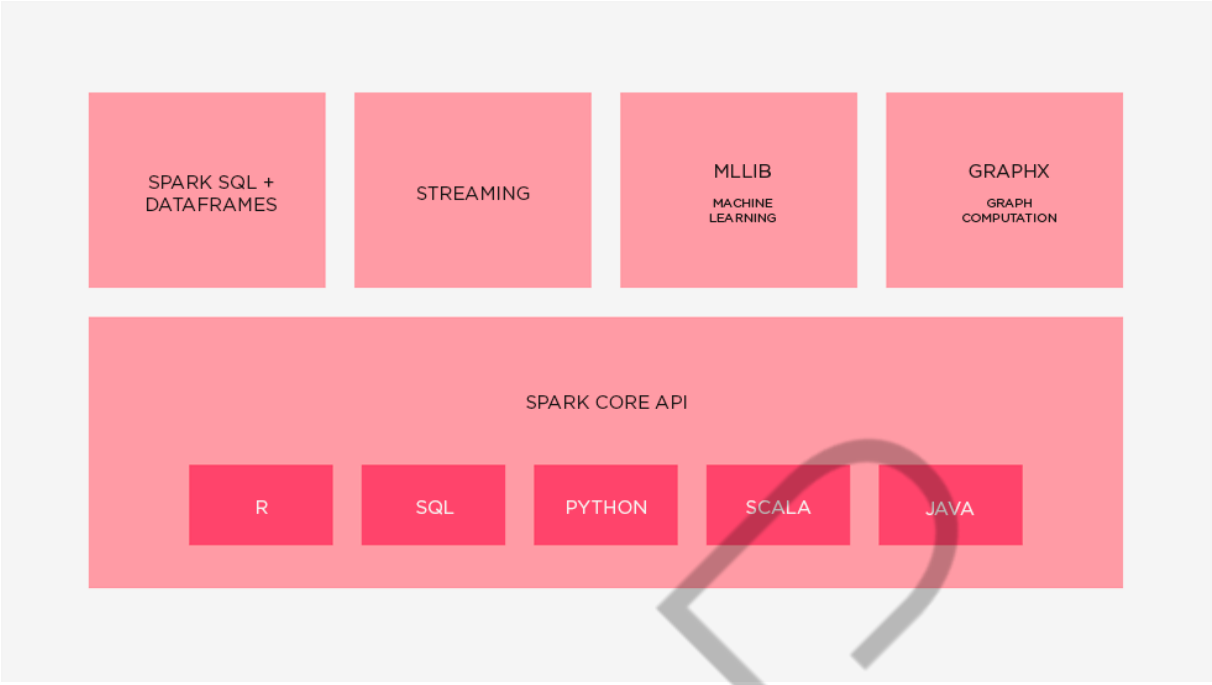


Figura 2 – Componentes do Apache Spark
Fonte: Mayara Machado (2020)

Vamos dar foco ao principal componente de SparkSQL, que é o **DataFrame.**

Ele é uma coleção de dados distribuídos e organizados em forma de colunas nomeadas. É baseado no conceito de estrutura de dados da linguagem R, similar a uma tabela de um banco de dados relacional.

Antes da versão 1.3, o componente DataFrame era chamado de SchemaRDD. DataFrames podem ser transformados em RDDs por meio de uma chamada de método RDD, que retorna o conteúdo de um DataFrame como um conjunto de linhas RDD.

DataFrames podem ser criados a partir de diferentes fontes de informações, como por exemplo:

- RDDs já existentes.
- Arquivos de dados estruturados.
- Conjunto de dados JSON.
- Tabelas Hive.
- Banco de dados externos.

O Spark SQL e a API de DataFrame estão disponíveis nas seguintes linguagens de programação:

- Scala
- Java
- Python

SQLCONTEXT

No Apache Spark, o SQLContext é uma interface que permite executar consultas e seleções de dados usando a linguagem SQL em conjuntos de dados distribuídos, como RDDs (Resilient Distributed Datasets) ou DataFrames. O SQLContext fornece uma maneira conveniente de trabalhar com dados estruturados usando a sintaxe familiar do SQL. Na figura 3 – “SQLContext”, temos um exemplo de como usar o SQLContext para consultas e seleções de dados no Spark.

```python
# Importando as bibliotecas necessárias
from pyspark.sql import SparkSession

# Criando uma SparkSession e um SQLContext
spark = SparkSession.builder.getOrCreate()
sqlContext = spark.sqlContext

# Carregando um arquivo CSV como DataFrame
df = spark.read.csv("caminho/do/arquivo.csv", header=True, inferSchema=True)

# Registrando o DataFrame como uma tabela temporária
df.createOrReplaceTempView("tabela_temporaria")

# Executando uma consulta SQL
resultado = sqlContext.sql("SELECT coluna1, coluna2 FROM tabela_temporaria WHERE coluna3 > 10")

# Exibindo o resultado
resultado.show()

# Encerrando a SparkSession
spark.stop()
```

Figura 3 – SQLContext
Fonte: Elaborado pelo autor (2023)

Neste exemplo, as etapas são as seguintes:

1. Importamos as bibliotecas necessárias, incluindo “SparkSession” para criar uma sessão Spark e “SQLContext” para acessar recursos SQL.

2. Criamos uma instância de “SparkSession” usando o “SparkSession.builder” e obtendo ou criando uma sessão existente.
3. A partir da sessão Spark, obtemos o “sqlContext” para acessar as funcionalidades do SQL.
4. Carregamos um arquivo CSV como um DataFrame usando o “spark.read.csv”, especificando o caminho do arquivo, indicando que a primeira linha é o cabeçalho e permitindo que o Spark inferisse o esquema dos dados.
5. Registramos o DataFrame como uma tabela temporária usando o “createOrReplaceTempView”, atribuindo um nome à tabela.
6. Executamos uma consulta SQL usando o “sqlContext.sql”, passando a consulta SQL como uma string.
7. Exibimos o resultado usando o “resultado.show()”, que mostra o resultado da consulta na saída.
8. Encerramos a SparkSession usando o “spark.stop()”.

O SQLContext fornece uma interface flexível para consultas e seleções de dados usando a sintaxe SQL padrão. Ele permite que você execute operações complexas em dados estruturados usando as funcionalidades do Spark SQL.

O QUE VOCÊ VIU NESTA AULA?

Em resumo, o Spark SQL é uma parte essencial do framework Apache Spark e fornece recursos poderosos para processar dados estruturados. Ele permite a execução de consultas SQL em diferentes fontes de dados, como arquivos batch, JSON e tabelas do HIVE, simplificando a análise e a manipulação de dados estruturados por meio da linguagem SQL.

Com o Spark SQL é possível realizar tarefas ETL, executar consultas ad-hoc e aproveitar a flexibilidade e a eficiência do Apache Spark para lidar com grandes volumes de dados.

REFERÊNCIAS

DOCUMENTAÇÃO Oficial Apache Spark. [s.d]. Disponível em https://spark.apache.org/documentation.html, [s.d.]. Acesso em: 10 jul. 2023.

LEARNING Spark. [s.d]. Disponível em: https://www.oreilly.com/library/view/learning-spark-2nd/9781492050032/ch04.html. Acesso em: 10 jul. 2023.

MACHADO, M. **Conhecendo o ecossistema Spark.** [s.d]. Disponível em: https://dev.to/mchdax/conhecendo-o-ecossistema-spark-1e54, 2020. Acesso em: 10 jul. 2023.

MYRIANTHOUS, G. **SparkSession vs SparkContext vs SQLContext**. 2020. Disponível em: https://towardsdatascience.com/sparksession-vs-sparkcontext-vs-sqlcontext-vs-hivecontext-741d50c9486a. Acesso em: 10 jul. 2023.

PENCHIKALA, S. **Big Data com Apache Spark Parte 2: Spark SQL**. 2016. Disponível em: https://www.infoq.com/br/articles/apache-spark-sql/. Acesso em: 10 jul. 2023.

RELVAS, C. **Apache Spark.** 2015. Disponível em: https://www.ime.usp.br/~gold/cursos/2015/MAC5742/reports/ApacheSpark.pdf. Acesso em 10 jul. 2023.

SPARK by Examples. [s.d]. Disponível em https://sparkbyexamples.com/spark/spark-sqlcontext-explained-with-examples/. Acesso em: 13 jul. 2023.

PALAVRAS-CHAVE

Apache Spark. Big Data. SQLContext. Hadoop. MapReduce. Big Data. Spark. Dados. Processamento.