UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Departamento Acadêmico de Informática

Big Data

Lab 3

Spark – Exemplos iniciais

Professor:	Leandro Batista de Almeida
Data:	3 de julho de 2025
Número de páginas:	6

Recursos:

- Referências
  - Spark site:
    - `https://spark.apache.org/`
  - Spark documentation site:
    - `https://spark.apache.org/docs/latest/`
  - PySpark Overview:
    - `https://spark.apache.org/docs/latest/api/python/index.html`

Ambiente da Atividade de Laboratório:

Esta atividade poderá ser realizada em um computador que tenha o Spark instalado, preferencialmente usando o sistema operacional Linux, mas também podendo ser executado em sistemas operacionais MacOS ou Windows.

Conteúdo:

1 Download e configuração do Spark

A partir do site do Spark, baixe a última versão estável (4.0, nesta data) com as opções default (Pre-built for Apache Hadoop).

Descompacte o arquivo baixado e mova o diretório criado para um local definitivo, por exemplo: “/home/leandro/spark-4.0” . Esse diretório será chamado de SPARK_HOME, de agora em diante.

Uma JDK deverá ser também instalado (versão 8, 11 ou 17), e o interpretador Python 3.x também deverá ser instalado.

2 Word Count Job em shell Scala

Baixe para um diretório do seu computador um arquivo TXT para a contagem de palavras, do Projeto Gutenberg ou outra fonte.

No diretório SPARK_HOME/bin, execute o script spark-shell.

Entre os seguintes comandos, substituindo o caminho e nome do arquivo pelo seu arquivo baixado, e também o caminho para o resultado.

Os comandos devem ser entrados no shell uma linha por vez, com a execução via ENTER.

Em spark-shell:

```
val texto = sc.textFile("/home/leandro/datasets/pg2600.txt")

val palavras = texto.flatMap(line => line.split(" "))

val contadores = palavras.map(palavra => (palavra,1)).reduceByKey(_ + _)

contadores.saveAsTextFile("/home/leandro/datasets/result")
```

3 Consultas sobre o CSV de Employees em PySpark

Para esta tarefa, baixe o arquivo compactado de Employees da página da disciplina no Moodle. Descompacte o arquivo em um diretório do seu computador.

Nesse arquivo são encontradas diversas versões do mesmo CSV, com tamanhos diferentes, você pode usar os menores para os testes e os maiores para validar o desempenho. Perceba que existem diversos arquivos, com tamanhos (e número de linhas) diferentes, mostrados pelo nome do arquivo. O arquivo employees.csv é o arquivo original, contendo aproximadamente 300 mil linhas de dados.

No diretório SPARK_HOME/bin, execute o script pyspark.

Digite os comandos abaixo, um por linha, alterando o caminho do arquivo de entrada (CSV) para o caminho do seu computador.

Em pyspark:

```
df = spark.read.csv("/home/leandro/datasets/employees.csv", header=True, sep=';')

df.show()
df.summary().show()

df_f = df.filter(df.gender == "F")
df_f.show()

df_m = df.where("gender = 'M'")
df_m.show()

df.createOrReplaceTempView("employees")
spark.sql("select * from employees where emp_no = 10002").show()
```

4 Uso do PySpark em Notebooks Jupyter ou IDEs

Para esta tarefa, instale em seu sistema operacional a aplicação Jupyter Notebook (https://jupyter.org/) ou uma IDE compatível.

A biblioteca PySpark também deverá ser instalada, o que pode ser feito com o comando pip (ou pip3).

Com essas ferramentas instaladas, crie um notebook no Jupyter e execute as mesmas operações do item anterior.

Incluir a biblioteca necessária e criar a SparkSession como indicado:

```
# importação da biblioteca PySpark / SparkSession
from pyspark.sql import SparkSession

# criação da SparkSession conectada à todos os núcleos da CPU local
spark = SparkSession.builder.appName('ex-pyspark').config('spark.master',
'local[*]').getOrCreate()
```

5 Acesso a DataFrames em Notebooks Jupyter

Carregar dataframes com os CSVs de employees e salaries.

Executar joins com funções e com SQL.