FIAP - FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA
DATA SCIENCE

Fábio Pereira de Lima – RM98803
Giovanna Cardoso Satorres – RM99944
Giullia Bianca Rocha Souza – RM552108
Gustavo Semenuk – RM550472
Mayara Prado Góes – RM98758

Sprint 2: Solução de Extração de Dados para o Challenge MinSait

BIG DATA ARCHITECTURE & DATA INTEGRATION

São Paulo
2023

★ PROCESSOS DE CARGA DE DADOS ENTRE O AMBIENTE LOCAL E O AMBIENTE HADOOP

A carga de dados entre o ambiente local e o ambiente Hadoop envolve processos distintos para a transferência de informações em ambas as direções.

Ao carregar dados do ambiente local para o ambiente Hadoop, é necessário realizar a preparação dos dados. Essa etapa envolve a formatação adequada, conversão de formatos, limpeza e tratamento de dados ausentes ou inconsistentes. Em seguida, os dados são divididos em blocos menores para aproveitar o processamento distribuído do Hadoop. Essa divisão é especialmente relevante quando lidamos com grandes volumes de dados. Após a divisão, é essencial escolher o formato de armazenamento mais adequado para os dados, como texto simples, CSV, JSON, Avro ou Parquet, entre outros. Por fim, os dados preparados e divididos são transferidos para o ambiente Hadoop. Essa transferência pode ser realizada utilizando o HDFS, o sistema de arquivos distribuído do Hadoop, ou por meio de ferramentas como Flume, Sqoop ou Nifi, desenvolvidas para facilitar a ingestão de dados no Hadoop.

Já a carga de dados do ambiente Hadoop para o ambiente local segue um fluxo inverso. Primeiramente, os dados desejados são selecionados para extração do ambiente Hadoop, podendo incluir arquivos específicos, diretórios ou resultados de consultas. Em seguida, ocorre a exportação desses dados do Hadoop para o ambiente local. Essa exportação pode ser feita por meio de ferramentas como Sqoop ou mediante o desenvolvimento de um programa personalizado para conectar-se ao Hadoop e transferir os dados. Caso seja necessário, é possível realizar a conversão de formatos para adequá-los ao ambiente local. Por fim, os dados convertidos podem ser transferidos do ambiente Hadoop para o ambiente local utilizando métodos convencionais de transferência de arquivos, como FTP, SCP ou ferramentas de sincronização de arquivos.

Esses processos de carga de dados permitem uma integração eficiente entre o ambiente local e o ambiente Hadoop, possibilitando o aproveitamento dos recursos de processamento e armazenamento distribuídos do Hadoop, bem como a extração dos resultados e análises em um ambiente familiar ao usuário.

★ ALTERNATIVAS PARA CARGA DE DADOS DO PROJETO COM FERRAMENTAS DO ECOSSISTEMA APACHE HADOOP

No nosso projeto, faremos a coleta de dados via planilhas CSV que serão preenchidas pelo usuário, além de bases de banco de dados públicos. Existem várias ferramentas disponíveis no ecossistema Apache Hadoop que podem facilitar a carga de dados do nosso projeto. Duas alternativas que condizem com as nossas necessidades são:

→ Apache Sqoop: O Apache Sqoop é uma ferramenta que tem como objetivo facilitar a importação de dados de sistemas de armazenamento de dados estruturados, como bancos de dados relacionais, para o Hadoop. Ele oferece recursos avançados de importação, como divisão automática de dados, paralelização e suporte a várias fontes de dados. No caso das planilhas CSV preenchidas pelo usuário, o Apache Sqoop pode ser usado para importar esses dados diretamente para o Hadoop, permitindo que possamos aproveitar os recursos e o poder de processamento do ecossistema para análises e processamentos futuros.

→ Apache Hive: O Apache Hive é uma ferramenta de análise de dados que permite consultar dados armazenados no Hadoop, que utiliza a linguagem SQL, tornando-a familiar para os usuários. A ferramenta nos permite definir esquemas e criar tabelas para organizar os dados das planilhas CSV e bases de dados públicas. A ferramenta aproveita o processamento distribuído do Hadoop, permitindo consultas e análises escaláveis de grandes volumes de dados. Além disso, integra-se facilmente com outras ferramentas do ecossistema Hadoop, como o Apache Sqoop. Ao utilizar o Hive, poderemos consultar, analisar e processar os dados de forma eficiente, facilitando a obtenção de insights valiosos.

★ CÓDIGOS

→ Apache Sqoop

Os códigos abaixo podem ser utilizados para carregar dados de um arquivo CSV e de um banco de dados MySQL para o Hadoop utilizando o Apache Sqoop.

— Importar dados de um arquivo CSV:

```
sqoop import \
--connect jdbc:csv:///path/to/csv/file.csv \
--table csv_table \
--target-dir /user/hadoop/csv_data \
--fields-terminated-by ','
```

- O comando **“SQOOP IMPORT”** é usado para importar dados de uma fonte externa para o Hadoop.
- O comando **“CONNECT JDBC:CSV:///PATH/TO/CSV/FILE.CSV”** especifica o caminho do arquivo CSV a ser importado. O prefixo **“JDBC:CSV://”** é usado para indicar que está lendo um arquivo CSV.
- O comando **“TABLE CSV_TABLE”** define o nome da tabela a ser criada no Hadoop para armazenar os dados importados do arquivo CSV.
- O comando **“TARGET-DIR /USER/HADOOP/CSV_DATA”** especifica o diretório de destino onde os dados importados serão armazenados no Hadoop.
- O comando **“FIELDS-TERMINATED-BY ','”** define o caractere de separação de campo no arquivo CSV. Nesse caso, está definido como ',' (vírgula).

— Importar dados de uma tabela SQL:

```
sqoop import \
--connect jdbc:mysql://localhost:3306/database_name \
--username mysql_user \
--password mysql_password \
--table mysql_table \
--target-dir /user/hadoop/mysql_data
```

- O comando **“SQOOP IMPORT”** é usado para importar dados de uma fonte externa para o Hadoop.
- O comando **“CONNECT JDBC:MYSQL://LOCALHOST:3306/DATABASE_NAME”** especifica o URL de conexão com o banco de dados MySQL que contém os dados a serem importados. O parâmetro **“LOCALHOST”** indica o endereço do servidor MySQL, 3306 é a porta padrão do MySQL e **“DATABASE_NAME”** é o nome do banco de dados.
- O comando **“USERNAME MYSQL_USER”** define o nome de usuário para autenticação no banco de dados MySQL.
- O comando **“PASSWORD MYSQL_PASSWORD”** especifica a senha para autenticação no banco de dados MySQL.
- O comando **“TABLE MYSQL_TABLE”** indica o nome da tabela no banco de dados MySQL que contém os dados a serem importados.
- O comando **“TARGET-DIR /USER/HADOOP/MYSQL_DATA”** define o diretório de destino no Hadoop onde os dados importados serão armazenados.

→ Apache Hive

Os códigos abaixo podem ser utilizados para carregar dados de um arquivo CSV e de um banco de dados MySQL para o Hadoop utilizando o Apache Hive.

— Importar dados de um arquivo CSV:

Primeiro, é preciso criar uma tabela no Hive que corresponda à estrutura dos dados no arquivo CSV.

```
CREATE TABLE csv_table (
  column1 data_type,
  column2 data_type,
  ...
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

- O comando **“CREATE TABLE”** é usado para criar uma tabela no Apache Hive.
- O comando **“CSV_TABLE”** é o nome da tabela que está sendo criada.
- O comando **“(COLUMN1 DATA_TYPE, COLUMN2 DATA_TYPE, ...)”** é a definição das colunas da tabela, onde deve-se listar cada coluna com seu respectivo nome e tipo de dados.
- O comando **“ROW FORMAT DELIMITED”** define o formato de linha dos dados na tabela.
- O comando **“FIELDS TERMINATED BY ','”** especifica o caractere utilizado para separar os campos em cada linha da tabela. Nesse caso, o caractere utilizado é a vírgula (',').
- O comando **“STORED AS TEXTFILE”** define o formato de armazenamento dos dados da tabela. Neste caso, os dados serão armazenados como arquivos de texto simples (formato TEXTFILE).

Em seguida, é preciso carregar os dados do arquivo CSV para a tabela no Hive.

```
LOAD DATA INPATH '/path/to/csv/file.csv' INTO TABLE csv_table;
```

- O comando SQL **“LOAD DATA INPATH”** é usado para carregar dados de um arquivo externo para uma tabela existente no Hive.
- O comando **“/PATH/TO/CSV/FILE.CSV”** especifica o caminho do arquivo CSV a ser carregado.
- O comando **“INTO TABLE CSV_TABLE”** indica a tabela de destino no Hive para a qual os dados do arquivo CSV serão carregados.

— Importar dados de uma tabela SQL:

Primeiro, é preciso criar uma tabela no Hive que corresponda à estrutura da tabela MySQL.

```
CREATE TABLE mysql_table (
  column1 data_type,
  column2 data_type,
  ...
)
STORED AS ORC;
```

- O comando SQL **“CREATE TABLE”** é usado para criar uma tabela no Apache Hive.
- O comando **“MYSQL_TABLE**” é o nome da tabela que está sendo criada.
- O comando **“(COLUMN1 DATA_TYPE, COLUMN2 DATA_TYPE, ...)”** é a definição das colunas da tabela, onde deve-se listar cada coluna com seu respectivo nome e tipo de dados.
- O comando **“STORED AS ORC”** especifica o formato de armazenamento dos dados da tabela. Neste caso, os dados serão armazenados no formato ORC (Optimized Row Columnar).

Em seguida, é necessário utilizar o Apache Sqoop para importar os dados da tabela MySQL para o Hive:

```
sqoop import \
--connect jdbc:mysql://localhost:3306/database_name \
--username mysql_user \
--password mysql_password \
--table mysql_table \
--hive-import \
--hive-table mysql_table \
--create-hive-table
```

- O comando **“SQOOP IMPORT”** é usado para importar dados de uma fonte externa para o Hadoop.
- O comando **“CONNECT JDBC:MYSQL://LOCALHOST:3306/DATABASE_NAME”** especifica o URL de conexão com o banco de dados MySQL que contém os dados a serem importados. O parâmetro **“LOCALHOST”** indica o endereço do servidor MySQL, 3306 é a porta padrão do MySQL e **“DATABASE_NAME”** é o nome do banco de dados.
- O comando **“USERNAME MYSQL_USE”** define o nome de usuário para autenticação no banco de dados MySQL.
- O comando password **“MYSQL_PASSWORD”** especifica a senha para autenticação no banco de dados MySQL.
- O comando table **“MYSQL_TABLE”** indica o nome da tabela no banco de dados MySQL que contém os dados a serem importados.
- O comando **“HIVE-IMPORT”** indica ao Sqoop que os dados devem ser importados para o Hive.
- O comando “**HIVE-TABLE MYSQL_TABLE**” especifica o nome da tabela no Hive onde os dados importados serão armazenados.
- O comando **“CREATE-HIVE-TABLE”** instrui o Sqoop a criar uma nova tabela no Hive, caso a tabela especificada pelo **--hive-table** não exista.