Trabalho - Big Data

Cassandra

Banco de dados

Criação do Keyspace

create keyspace ks_nasa with replication =
{'class':'SimpleStrategy','replication_factor':1};

Criando a tabela

use ks_nasa;
create table tbl_sensors (id TIMEUUID primary key, sensor_name text, value text);

Depois

create table tbl_sensors (id TIMEUUID primary key, created_at text, sensor_0 text, sensor_1 text, sensor_2 text, sensor_3 text, sensor_4 text, sensor_5 text, sensor_6 text, sensor_7 text, sensor_8 text, sensor_9 text, sensor_10 text, sensor_11 text, sensor_12 text, sensor_13 text, sensor_14 text, sensor_15 text, sensor_16 text, sensor_17 text, sensor_18 text, sensor_19 text, sensor_20 text);

Depois

create table tbl_sensors (id TIMEUUID primary key, created_at text, file_type text, unit_number text, cycles text, setting1 text, setting2 text, setting3 text, sensor_1 text, sensor_2 text, sensor_3 text, sensor_4 text, sensor_5 text, sensor_6 text, sensor_7 text, sensor_8 text, sensor_9 text, sensor_10 text, sensor_11 text, sensor_12 text, sensor_13 text, sensor_14 text, sensor_15 text, sensor_16 text, sensor_17 text, sensor_18 text, sensor_19 text, sensor_20 text, sensor_21 text);

Editando o arquivo cassandra-sink-standalone.properties

topics=topic_sensors

topic.topic_sensors.ks_nasa.tbl_sensors.mapping=id=now(), sensor_name=value.sensor_name, value=value.value

Depois

topic.topic_sensors.ks_nasa.tbl_sensors.mapping=id=now(), created_at=value.created_at, sensor_0=value.sensor_0, sensor_1=value.sensor_1 , sensor_2=value.sensor_2 , sensor_3=value.sensor_3 , sensor_4=value.sensor_4 , sensor_5=value.sensor_5 , sensor_6=value.sensor_6 , sensor_7=value.sensor_7 , sensor_8=value.sensor_8 ,

sensor_9=value.sensor_9 , sensor_10=value.sensor_10 ,
sensor_11=value.sensor_11 , sensor_12=value.sensor_12 ,
sensor_13=value.sensor_13 , sensor_14=value.sensor_14 ,
sensor_15=value.sensor_15 , sensor_16=value.sensor_16 ,
sensor_17=value.sensor_17 , sensor_18=value.sensor_18 ,
sensor_19=value.sensor_19 , sensor_20=value.sensor_20

Depois

topic.topic_sensors.ks_nasa.tbl_sensors.mapping=id=now(),
created_at=value.created_at, file_type=value.file_type,
unit_number=value.unit_number, cycles=value.cycles, setting1=value.setting1,
setting2=value.setting2, setting3=value.setting3, sensor_1=value.sensor_1,
sensor_2=value.sensor_2, sensor_3=value.sensor_3, sensor_4=value.sensor_4,
sensor_5=value.sensor_5, sensor_6=value.sensor_6, sensor_7=value.sensor_7,
sensor_8=value.sensor_8, sensor_9=value.sensor_9,
sensor_10=value.sensor_10, sensor_11=value.sensor_11,
sensor_12=value.sensor_12, sensor_13=value.sensor_13,
sensor_14=value.sensor_14, sensor_15=value.sensor_15,
sensor_16=value.sensor_16, sensor_17=value.sensor_17,
sensor_18=value.sensor_18, sensor_19=value.sensor_19,
sensor_20=value.sensor_20, sensor_21=value.sensor_21

topic.topic_sensors.ks_nasa.tbl_sensors.ttlTimeUnit=SECONDS
topic.topic_sensors.ks_nasa.tbl_sensors.timestampTimeUnit=MICROSECONDS

Iniciando os serviços

1. Iniciar o Zookeeper (zookeeper-server-start /opt/homebrew/etc/kafka/zookeeper.properties)
2. Iniciar o Kafka (rm -rf /opt/homebrew/var/lib/kafka-logs && kafka-server-start /opt/homebrew/etc/kafka/server.properties)
3. Resetar o topico (kafka-topics --bootstrap-server localhost:9092 --topic topic_sensors --delete)
4. Iniciar o Cassandra
5. Copiar o JAR do conector para o Kafka
6. Iniciar o connect-standalone (/opt/homebrew/Cellar/kafka/3.4.0/bin/connect-standalone /opt/homebrew/etc/kafka/connect-standalone.properties /opt/homebrew/etc/kafka/cassandra-sink-standalone.properties)

Programando

Decidimos usar apenas um dos arquivos do dataset para criar o código, o arquivo usado foi o "test_FD003.txt".

Producers

Definimos que, como cada coluna representa um sensor, vamos tornar cada coluna um Producer que vai enviar apenas os dados do seu sensor.
Como as linhas de 0 a 4 são configurações e não valores de sensores não serão enviados como um Producer. (Portanto, 5 primeiras colunas serão removidas)

Consumer
Vai receber os dados dos sensores e apenas salvar no Cassandra.

Passos

- ✅ Ler o dataset como streams para simular um funcionamento continuo
- ✅ Criar um Producer para cada sensor
- ✅ Enviar os dados no tópico “topic_sensors”
- ✅ Criar o consumer que vai pintar o nome do sensor, o timestam e o valor recebido
- ✅ Consultar no “cqlsh” os valores inseridos

Resultado (Controlado)

```
cqlsh:ks_nasa> select * from tbl_sensors;

 id                                   | sensor_name | value
--------------------------------------+-------------+---------
 36446763-ec1e-11ed-8959-d9dd341e2ef1 |    sensor_8 | 9048.65
 364503a9-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_16 |     391
 364503a5-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_19 |   39.07
 364503a2-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_14 |  8.3760
 36446765-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_10 |   47.09
 364503a4-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_20 | 23.4468
 36444052-ec1e-11ed-8959-d9dd341e2ef1 |    sensor_4 |   14.62
 36446762-ec1e-11ed-8959-d9dd341e2ef1 |    sensor_9 |    1.30
 363f1030-ec1e-11ed-8959-d9dd341e2ef1 |    sensor_1 |  641.94
 364503a3-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_13 | 8133.48
 364503a0-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_11 |  521.89
 36446764-ec1e-11ed-8959-d9dd341e2ef1 |    sensor_7 | 2387.93
 364503a6-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_18 |  100.00
 364503a8-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_15 |    0.03
 364503a1-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_12 | 2387.94
 36446760-ec1e-11ed-8959-d9dd341e2ef1 |    sensor_5 |   21.58
 36444051-ec1e-11ed-8959-d9dd341e2ef1 |    sensor_3 | 1396.93
 36446761-ec1e-11ed-8959-d9dd341e2ef1 |    sensor_6 |  554.56
 36444050-ec1e-11ed-8959-d9dd341e2ef1 |    sensor_2 | 1581.93
 364503a7-ec1e-11ed-8959-d9dd341e2ef1 |   sensor_17 |    2388
 363f1031-ec1e-11ed-8959-d9dd341e2ef1 |    sensor_0 |  518.67
```

Para filtrar no cassandra usamos:
SELECT * FROM tbl_sensors WHERE file_type = 'train' ALLOW FILTERING;
SELECT * FROM tbl_sensors WHERE file_type = ‘test’ ALLOW FILTERING;