Processamento de stream em Apache Kafka

Nesta atividade utilizou-se o programa Apache Kafka para produzir e consumir um stream de dados provindos do site “Wikipedia”. Nesse stream de dados contém dados brutos sobre modificações feitas na página do site e a proposta da atividade é extrair informações desejadas e repassá-las a um tópico. Para tanto seguiu-se a seguinte proposta:

- Criou-se um tópico chamado “wikiUsuariosMQ”

```
1. sh kafka-topics.sh --create --zookeeper localhost:2181 --if-not-exists --replication-factor 1 --partitions 1 --topic wikiUsuariosMQ --config retention.ms=1000
```

```
[godoi.felipe@ead-bigdata01-docker bin]$ sh kafka-topics.sh --create --zookeeper l
ocalhost:2181 --if-not-exists  --replication-factor 1 --partitions 1 --topic wikiU
suariosMQ --config retention.ms=1000
Created topic "wikiUsuariosMQ".
```

Figura 1 - Tópico "wikiUsuarioMQ" criado

- Criou-se um consumidor para o stream “wikiMQ”

```
2. /kafka-console-consumer.sh --bootstrap-server 172.18.0.2:6667 --topic wikiMQ --from-beginning --group wiki
```

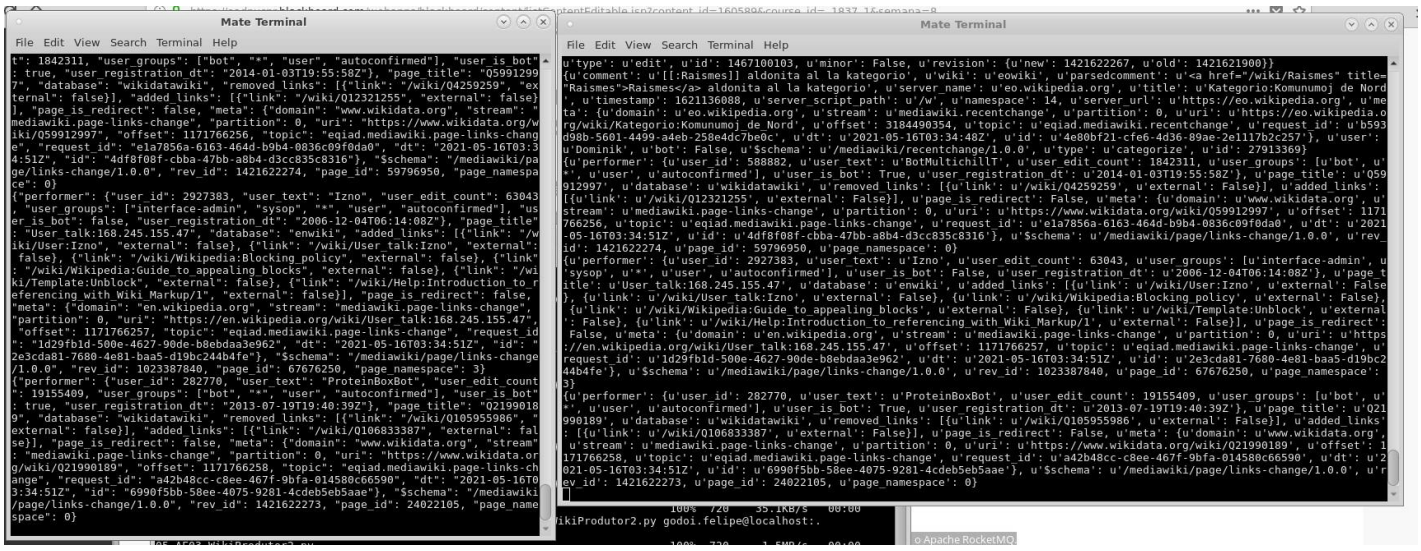

Figura 2 - Produtor e consumidor em funcionamento

- A partir de um script feito em Python, iniciou-se um stream de dados para o tópico “wikiMQ” contendo os dados obtidos do site wikipédia

```
import json
```

```
from kafka import SimpleProducer, KafkaClient
from sseclient import SSEClient as EventSource

kafka = KafkaClient("172.18.0.2:6667")
producer = SimpleProducer(kafka)

streams = 'recentchange,page-links-change,page-create,page-move,page-
properties-change,page-delete,test,recentchange,revision-create,page-undelete'
url = 'https://stream.wikimedia.org/v2/stream/{}'.format(streams)

for event in EventSource(url):
    try:
        if event.event == 'message':
            change = json.loads(event.data)
            print (change)
            producer.send_messages("wikiMQ", json.dumps(change))
    except Exception as e:
        print(e)
```

```
TOPIC                PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG  CONSUMER-ID                                    HOST            CLIENT-ID
wikiMQ               0          25194           25357           163  consumer-1-783f95fb-d7a5-46fb-a709-22695b5650a6  /172.18.0.2     consumer-1
WikiMQ               0          0               0               0    -                                              -               -
wikiMQav             0          0               0               0    -                                              -               -
```

Figura 4- Fila do tópico wikiMQ

```
File  Edit  View  Search  Terminal  Help
1621268178,DPLA bot
1621268176,NewsBots
1621268178,Foscolo
1621268176,BARBARE42
1621268172,NewsBots
1621268181,Mr.Ibrahembot
1621268173,Evelym-rose
1621268173,Sixfish
1621268180,Amasuela
1621268178,Foscolo
1621268172,NewsBots
1621268171,GeographBot
1621268164,NewsBots
1621268178,Foscolo
1621268172,NewsBots
1621268179,Trimton
1621268181,NewsBots
1621268176,Taamu
1621268181,Mr.Ibrahembot
1621268172,NewsBots
1621268180,NewsBots
1621268174,NewsBots
1621268178,GeographBot
1621268172,NewsBots
1621268168,NewsBots
1621268180,Amasuela
1621268164,NewsBots
1621268172,NewsBots
1621268180,ProteinBoxBot
1621268172,NewsBots
1621268165,NewsBots
1621268181,Albedo
1621268181,Omotecho
1621268181,Peter James
1621268172,NewsBots
```

Figura 3 - Consumidor genérico do tópico wikiMQ

- A partir do stream de dados contidos no tópico “wikiMQ”, criou-se outro script em Python para extrair desses dados a “timestamp” e o nome usuário que realizou uma modificação no site wikipedia e inseriu essas informações ao tópico “wikiUsuariosMQ”

```
import json
from kafka import KafkaConsumer, SimpleProducer, KafkaClient

kafka = KafkaClient("172.18.0.2:6667")
producer = SimpleProducer(kafka)

consumer = KafkaConsumer(
    'wikiMQ',
     bootstrap_servers=['172.18.0.2:6667'],
     auto_offset_reset='earliest',
     enable_auto_commit=True,
     group_id='wiki')


for message in consumer:
    try:
        jsonList = json.loads(message.value)
        strData = str(jsonList["timestamp"]) + "," + jsonList["user"]
        print(strData)
        producer.send_messages("wikiUsuariosMQ", strData.encode('utf-8'))
    except:
        pass
```

- Por último, criou-se um consumidor genérico para ler a fila do tópico “wikiUsuarioMQ” e testar a aplicação

```
3.  ./kafka-console-consumer.sh --bootstrap-server 172.18.0.2:6667 --topic wikiUsuariosMQ --
    from-beginning --group wiki
```

```
Note: This will not show information about old Zookeeper-based consumers.


TOPIC                          PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG        CONSUMER-ID                                    HOST                           CLIENT-ID
wikiMQ                         0          67512           67865           353        kafka-python-1.4.7-a6a931ca-886f-4fba-b69c-4837143e1a28/172.18.0.2  kafka-python-1.4.7
wikiUsuariosMQ                 0          14445           14591           146        consumer-1-0d5ab50c-3d60-422d-ac77-a1422340fbd4  /172.18.0.2                    consumer-1
WikiMQ                         0          0               0               0          -                                              -                              -
wikiMQav                       0          0               0               0          -                                              -                              -
```

Figura 5 - Fila para o tópico wikiUsuarioMQ

Figura 6 - Consumidor genérico do tópico "wikiUsuarioMQ"

Conclusão

Houve um pouco de dificuldade a respeito da adaptação dos scripts repassados para a atividade e na extração dos valores, mas estudando as ferramentas dispostas, formato dos dados e com algumas tentativas foi possível contorná-las, atingindo os objetivos do trabalho. tanto os produtores quanto os consumidores agiram conforme esperado a respeito dos dados enviados